



M 2015

# **CONTROLO VISUAL DE POSIÇÃO DE EMBARCAÇÃO AUTÓNOMA**

**MAURÍCIO MIGUEL DE OLIVEIRA GUEDES**

DISSERTAÇÃO DE MESTRADO APRESENTADA

À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM

MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

A Dissertação intitulada

“Controlo Visual de Posição de Embarcação Autónoma”

foi aprovada em provas realizadas em 17-07-2015


o júri



Presidente Professor Doutor Mário Jorge Rodrigues de Sousa  
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores  
da Faculdade de Engenharia da Universidade do Porto



Professor Doutor José Luís Magalhães Lima  
Professor Adjunto do Departamento de Eletrotécnica da Escola Superior de  
Tecnologia e Gestão do Instituto Politécnico de Bragança



Professor Doutor Aníbal Castilho Coimbra de Matos  
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores  
da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Maurício Miguel de Oliveira Guedes



**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



# **Controlo Visual de Posição de Embarcação Autónoma**

**Maurício Miguel de Oliveira Guedes**

**MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES**

**Orientador: Prof. Dr. Aníbal Castilho Coimbra de Matos**

**30 de Julho de 2015**



# Resumo

A vídeo-vigilância em cenas dinâmicas é atualmente um dos tópicos mais ativos no mundo da computação visual. A monitorização e vigilância de embarcações é deveras importante para a segurança e defesa marítima, proteção do ambiente e controlo de fronteiras marítimas. No entanto, a grande maioria dos sistemas desenvolvidos utiliza sistemas de visão conjugados com radares, AIS ou VTS. Nesta dissertação será apresentado um modelo de um detetor automático de embarcações com vista a criar um registo de todas as ocorrências marítimas com o propósito de criar um controlador de tráfego marítimo autónomo. O detetor foi desenvolvido através de métodos de reconhecimento de padrões, originalmente desenvolvidos no campo da visão computacional. Primeiramente, é feito um treino de um classificador binário a partir de imagens de embarcações e de planos de fundo. O classificador utiliza características simples que são calculadas extremamente rápido. Para a deteção, este classificador percorre a imagem em várias direções e dimensões e tem uma estrutura em cascata o qual rejeita a maior parte dos planos de fundo nas etapas iniciais, levando a uma execução mais rápida. Antes da execução do detetor, é aplicado um corte na imagem através da deteção da linha do horizonte, retendo apenas a parte da imagem abaixo da linha. Durante a aplicação do detetor, são registadas todas as embarcações detetadas, as quais sofrem uma posterior validação através de um mapa temporal adaptativo, que regista as ocorrências passadas. É apresentada também uma modelização para transformação de coordenadas 2D do plano da imagem para coordenadas 3D no referencial do mundo, podendo desta forma obter informações posicionais das embarcações detetadas, com recurso a sensores de GPS e IMU instalados no sistema de aquisição de imagem.



# Abstract

Visual-surveillance in dynamic scenes is, currently, one of the most active research topics in the computer-vision field. Vessel monitoring and surveillance is important to the maritime safety and security, environment protection and maritime border control. However, a vast majority of existing systems combines visual systems with radar, AIS or VTS. In this dissertation, we will present a model of an automatic vessel detector for the purpose of ship traffic management. This vessel detector method uses pattern recognition methods, originally developed in the computer vision field. First, we train a binary classifier from image samples of vessels and background. The classifier uses simple features that are calculated very fast. For the detection, the detector is slid along the image, in various directions and scales. The classifier has a cascade structure which rejects most of the background in the early stages, leading to faster execution. Before running the detector, the image is cropped after the detection of the horizon line, retrieving only the part of the image below the horizon line. During the execution of the detector, all detected vessels are registered and validated using an adaptive time map that records past occurrences in detection. We present also a model for converting 2D coordinates, in the image plane, into 3D coordinates on the world coordinate system, with the possibility of retrieving positional information about the detected vessels, by means of GPS and IMU sensors installed on the vision system.





# Agradecimentos

Gostaria de expressar a minha profunda e sincera gratidão aos meus pais pelos valores que me inculcaram, pelo incentivo e apoio desde sempre, para que me tornasse uma pessoa melhor e mais sábia e por todo os esforços que fizeram para que eu hoje pudesse ser quem sou. A vós, o meu muito Obrigado!

Ao meu tio, Nuno Guedes, por me ter apoiado desde sempre e me ter demonstrado a essência de ser um engenheiro, Obrigado!

Aos restantes membros da minha família, que sempre me apoiaram e me incentivaram a seguir em frente na conquista de novos desafios, na perseverança e sempre com uma atitude positiva, Obrigado!

À minha namorada, Matilde, que se mostrou sempre disponível em contribuir para que eu chegasse tão longe, que esteve sempre ao meu lado, me apoiou e motivou, Obrigado!

Aos meus companheiros e amigos que, direta ou indiretamente, me acompanharam e ajudaram neste caminho. Obrigado por todos os momentos que passámos.

Por fim, gostaria de agradecer ao meu orientador Prof. Dr. Aníbal Castilho Coimbra de Matos pela oportunidade que me proporcionou ao realizar este projeto, por todo o conhecimento transmitido em todo o meu percurso académico e pela disponibilidade, compreensão e incentivo demonstrados no decorrer desta dissertação.

A todos Muito Obrigado,

Maurício Miguel de Oliveira Guedes



*“We live on an island surrounded by a sea of ignorance.  
As our island of knowledge grows, so does the shore of our ignorance.”*

John Archibald Wheeler



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objectivos . . . . .	2
1.3	Estrutura do Documento . . . . .	2
<b>2</b>	<b>Estado da Arte</b>	<b>5</b>
2.1	Veículos não tripulados . . . . .	5
2.1.1	Embarcações Autónomas . . . . .	6
2.2	Sistemas de Visão . . . . .	7
2.2.1	Tipos de Sistemas de Visão . . . . .	8
2.3	Técnicas de Detecção de Objetos . . . . .	9
2.3.1	Radar . . . . .	9
2.3.2	Subtração de Fundo . . . . .	9
2.3.3	Subtração de Fundo Adaptativa . . . . .	10
2.3.4	Classificadores . . . . .	10
2.4	Trabalhos relacionados . . . . .	11
<b>3</b>	<b>Abordagem ao problema</b>	<b>15</b>
3.1	Formulação do Problema . . . . .	15
3.2	Solução Proposta . . . . .	16
3.2.1	Arquitectura funcional da solução . . . . .	16
3.2.2	Descrição do Sistema . . . . .	16
3.3	Síntese . . . . .	17
<b>4</b>	<b>Modelização da Solução Proposta</b>	<b>19</b>
4.1	Geometria Projetiva . . . . .	19
4.2	Modelo de uma Câmara <i>pin-hole</i> . . . . .	20
4.2.1	Parâmetros Intrínsecos da Câmara . . . . .	20
4.2.2	Parâmetros Extrínsecos da Câmara . . . . .	25
4.2.3	Projeção de um ponto 2D em 3D . . . . .	25
4.2.4	Calibração da Câmara . . . . .	26
4.3	Aquisição . . . . .	28
4.4	Processamento . . . . .	29
4.4.1	Entrada de Vídeo ou <i>Stream</i> . . . . .	29
4.5	Pós-Processamento . . . . .	30
4.6	Registo e Tratamento da Informação . . . . .	31
4.7	Biblioteca OpenCv . . . . .	31

<b>5</b>	<b>Aquisição dos Dados de Visão e Posicionais</b>	<b>33</b>
5.1	Aquisição de Vídeos . . . . .	33
5.2	Estação de Aquisição . . . . .	34
5.2.1	Módulo <i>Arduino</i> e Comunicação . . . . .	35
5.2.2	Aquisição do Sinal do GPS . . . . .	35
5.2.3	Rotação do sistema de aquisição . . . . .	35
5.3	Discussão de Resultados e Conclusão . . . . .	37
<b>6</b>	<b>Processamento de Imagem</b>	<b>39</b>
6.1	Deteção da Linha do Horizonte e Região de Interesse . . . . .	39
6.2	Classificador em cascata com características <i>Haar</i> . . . . .	42
6.2.1	Características <i>Haar</i> . . . . .	43
6.2.2	<i>AdaBoost</i> . . . . .	45
6.2.3	Classificadores em Cascata . . . . .	46
6.3	Treino do Classificador . . . . .	48
6.3.1	Positivos . . . . .	48
6.3.2	Negativos . . . . .	50
6.3.3	Aprendizagem . . . . .	51
6.4	Resultados da Deteção . . . . .	53
6.5	Mapa Temporal . . . . .	56
<b>7</b>	<b>Conclusões</b>	<b>59</b>
7.1	Conclusões . . . . .	59
7.2	Desenvolvimentos Futuros . . . . .	60
	<b>Referências</b>	<b>61</b>



# Lista de Figuras

2.1	<i>Exemplo de um USV com casco de catamarã (figura retirada de [1])</i>	6
2.2	<i>FAST, Veleiro Autônomo da FEUP (figura retirada de [2])</i>	7
2.3	<i>Embarcação autônoma C-Worker para operações marítimas (figura retirada de [3])</i>	8
2.4	<i>Embarcação autônoma C-Enduro para recolher dados em alto mar (figura retirada de [4])</i>	8
2.5	<i>Diagrama da técnica de subtração de fundo.</i>	10
2.6	<i>Exemplo da subtração de fundo adaptativa ViBe.</i>	10
2.7	<i>Exemplo da determinação de características Haar em várias imagens.</i>	11
2.8	<i>Ilustração do sistema de radar TerraSAR-X.</i>	12
2.9	<i>Aquisição de imagem por um sistema visual infravermelho (figura retirada de [5])</i>	12
3.1	<i>Ilustração da solução proposta.</i>	16
3.2	<i>Arquitetura funcional da solução proposta</i>	17
4.1	<i>Duas retas paralelas interseitam-se no infinito, no ponto de fuga (figura retirada de [6]).</i>	19
4.2	<i>O modelo ideal de uma câmara pin-hole descreve a relação entre um ponto 3D <math>(X,Y,Z)^T</math> e a sua respetiva projeção 2D <math>(U,V)^T</math> no plano da imagem.</i>	21
4.3	<i>Os referenciais da imagem <math>(x,y)</math> e da câmara <math>(u,v)</math>.</i>	22
4.4	<i>Sensor de imagem não ideal com pixeis desviados e não quadrados.</i>	23
4.5	<i>As lentes das câmaras sofrem distorção radial. Grelha de pixeis de uma imagem não distorcida 4.5a e distorcida 4.5b</i>	24
4.6	<i>A relação entre o referencial da câmara e o do Mundo é definida pelo centro da câmara C e pela rotação R da câmara.</i>	25
4.7	<i>Estimação das coordenadas de uma embarcação no ponto P, através da projeção das suas coordenadas nos eixos alinhados com Norte e Oeste.</i>	27
4.8	<i>Exemplo de uma câmara PTZ (figura retirada de [7]).</i>	28
4.9	<i>Eixos de rotação da câmara.</i>	29
4.10	<i>Diagrama do algoritmo de aquisição de imagem.</i>	30
4.11	<i>Deteção de 2 "embarcações" numa imagem, como dois retângulos, e respetivas coordenadas e dimensões.</i>	31
5.1	<i>Câmara utilizada na aquisição dos vídeos de teste.</i>	33
5.2	<i>Exemplos de imagens captadas em diversos vídeos.</i>	34
5.3	<i>Protótipo de uma estação de aquisição.</i>	34
5.4	<i>Módulo de GPS implementado na estação de aquisição (figura retirada de [8]).</i>	36
5.5	<i>Módulo de GPS implementado na estação de aquisição.</i>	36
5.6	<i>Servo-motor utilizado na estação de aquisição para variação da pose da câmara (figura retirada de [9]).</i>	36

5.7	<i>Teste de funcionamento da estação de aquisição e aquisição do sinal de GPS.</i>	37
6.1	<i>Aplicação do limiar por método de Otsu.</i>	40
6.2	<i>Aplicação do detetor de pixeis de fronteira Canny</i>	40
6.3	<i>Fotogramas analisados e linhas detetadas após transformada de Hough.</i>	40
6.4	<i>Limiarização por método de Otsu à esquerda e transformada de Hough à direita.</i>	41
6.5	<i>Limiar por método de Otsu à esquerda e transformada de Hough à direita após limiar adaptativo.</i>	41
6.6	<i>Validação da linha em análise, a amarelo, e linha do horizonte, após validação, a verde.</i>	42
6.7	<i>Região de interesse retirada da imagem original.</i>	42
6.8	<i>Caraterísticas Haar utilizadas.</i>	43
6.9	<i>Áreas da soma de imagens integrais.</i>	44
6.10	<i>Cálculo da área sombreada de uma imagem integral.</i>	45
6.11	<i>As primeiras 3 etapas de um classificador em cascata. Cada etapa contém várias árvores de classificação simples que utilizam as características Haar e um limiar "thr" e retornam <math>\sum h(x)</math>, determinado através do percurso pelas árvores.</i>	47
6.12	<i>Exemplos positivos de embarcações.</i>	49
6.13	<i>Marcação de uma embarcação num exemplo positivo.</i>	49
6.14	<i>Alguns exemplos negativos utilizados.</i>	50
6.15	<i>Treino de um classificador com 16 etapas (6.15a) e de um classificador com 20 etapas (6.15b).</i>	51
6.16	<i>Última de 20 etapas de um classificador com taxa de falsos positivos = 0,4, utilizando para isso 43 classificadores fracos.</i>	52
6.17	<i>Classificador em cascata com 16 etapas.</i>	53
6.18	<i>Classificador em cascata com 18 etapas.</i>	54
6.19	<i>Classificador em cascata com 20 etapas.</i>	54
6.20	<i>Gráfico de deteções com um classificador em cascata com 16 etapas para o vídeo 00127.MTS com 3 embarcações.</i>	54
6.21	<i>Gráfico de deteções com um classificador em cascata com 18 etapas para o vídeo 00127.MTS com 3 embarcações.</i>	54
6.22	<i>Gráfico de deteções com um classificador em cascata com 20 etapas para o vídeo 00127.MTS com 3 embarcações.</i>	54
6.23	<i>Gráfico de deteções com um classificador em cascata com 24 etapas para o vídeo 00127.MTS com 3 embarcações.</i>	55
6.24	<i>Gráfico de deteções com um classificador em cascata com 20 etapas e imagens equalizadas, para o vídeo 00127.MTS com 3 embarcações.</i>	56
6.25	<i>Gráfico de deteções com um classificador em cascata com 24 etapas e imagens equalizadas, para o vídeo 00127.MTS com 3 embarcações.</i>	56
6.26	<i>Deteção e Validação em execução.</i>	57
6.27	<i>Mapa temporal.</i>	57
6.28	<i>Exemplo de uma deteção errada descartada pelo mapa temporal.</i>	58

# Lista de Tabelas

6.1	<i>Tabela de comparação entre os classificadores treinados, para 200 fotogramas.</i>	55
6.2	<i>Tabela de comparação entre os classificadores treinados, com imagens equalizadas, para 200 fotogramas.</i>	55



# Abreviaturas e Símbolos

2D	Bidimensional (2 Dimensional)
3D	Tridimensional (3 Dimensional)
AIS	Sistema Automático de Identificação (Automatic Identification System)
FPGA	Arranjo de Portas Programável em Campo(Field-Programmable Gate Array)
$I^2C$	Circuito Inter-Integrado (Inter-Integrated Circuit)
IMU	Unidade de Medição Inercial (Inercial Measuring Unit)
GPS	Sistema de Posicionamento Global (Global Positioning System)
PTZ	Panorâmica-Inclinação-Zoom (Pan-Tilt-Zoom)
PWM	Modulação por Largura de Impulso (Pulse-Width Modulation)
RTSP	Protocolo de Fluxo em Tempo Real (Real Time Streaming Protocol)
UAV	Veículo Aéreo Não Tripulado (Unmanned Aerial Vehicle)
UDP	Protocolo de Datagramas do Utilizador (User Datagram Protocol)
UGV	Veículo Terrestre Não Tripulado (Unmanned Ground Vehicle)
USB	Universal Serial Bus
USV	Veículo de Superfície Não Tripulado (Unmanned Surface Vehicle)
UUV	Veículo Submarino Não Tripulado (Unmanned Underwater Vehicle)
VTs	Serviço de Tráfego Marítimo (Vessel Traffic Service)
WGS 84	Sistema Geodésico Mundial 1984 (World Geodetic System 1984)





# Capítulo 1

## Introdução

Este projeto tem como objetivo a deteção, monitorização e seguimento de embarcações autónomas aquáticas em ambiente marítimo não exclusivo, através de um sistema de visão computacional, de forma a obter informações respeitantes às mesmas, como distância, posição no mundo, velocidade, trajetória e dimensão.

Este documento apresenta todo o trabalho efetuado para a unidade curricular Dissertação, no âmbito do Mestrado Integrado de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto. Desta forma, terá apresentada a Motivação ( 1.1) para a elaboração da respetiva dissertação, seguida dos Objetivos ( 1.2) alvo deste projeto e, por fim, a respetiva Estrutura ( 1.3) do presente documento.

### 1.1 Motivação

A utilização de veículos não tripulados cada vez mais proporciona um amplo leque de vantagens, seja na redução de risco em tarefas perigosas para os humanos, redução de custos de desenvolvimento e operabilidade, acesso a áreas remotas, assim como melhor eficiência energética, dada a ausência presencial de um controlador humano. Desta forma, torna-se muito interessante a incorporação deste tipo de veículos em diversos sistemas atuais. Esta simbiose pode já ser observada em aplicações militares de alto nível, como é o caso dos veículos aéreos não tripulados, bem como em explorações marítimas profundas e ultra-profundas e até mesmo em explorações espaciais.

Assim, e enfatizando os veículos autónomos, torna-se imperativo determinar e controlar a posição destes veículos de forma a maximizar a sua fiabilidade aquando da sua incorporação em espaços não exclusivos. Através de uma combinação de informações externas ao veículo com dados de bordo, é possível fazer correções e modificações aos seus algoritmos de controlo que permitam garantir o correto funcionamento deste em qualquer situação. Da mesma forma, é também possível fazer uma gestão do tráfego destes veículos através do planeamento e controlo em tempo real das movimentações dos mesmos, assegurando a segurança da circulação e maximização da eficiência dos veículos, tanto energética como na tomada de decisões de trajetória.

Esta dissertação abordará estes temas respeitante a embarcações marítimas e combinará um sistema de visão estacionário com informações de bordo da embarcação, esperando criar um sistema fiável e útil, num futuro breve, para controlo da posição de embarcações autónomas em espaços não exclusivos e limitados.

## 1.2 Objectivos

Os objetivos propostos para esta dissertação são:

1. Estudo da relação entre um ponto no espaço 3D e a sua respetiva projeção 2D;
2. Estudo da função de projeção de coordenadas 2D num sistema de coordenadas 3D;
3. Estudo e desenvolvimento dos algoritmos de visão para processamento de imagem;
4. Estudo e desenvolvimento do algoritmo de deteção da linha do horizonte;
5. Estudo e desenvolvimento do algoritmo de visão para deteção de embarcações;
6. Implementação, teste e validação dos algoritmos desenvolvidos, em vídeos;
7. Comparação dos vários detetores desenvolvidos;
8. Desenvolvimento do algoritmo de validação das embarcações detetadas na imagem;
9. Desenvolvimento de uma estação de aquisição e transmissão de dados de imagem e dados posicionais.

## 1.3 Estrutura do Documento

Além da introdução, este documento encontra-se dividido em mais 6 capítulos.

No capítulo 2, é descrito o estado da arte relativo às tecnologias atualmente existentes na área dos veículos não tripulados e dos sistemas de visão existentes, acoplados ou não.

No capítulo 3, são apresentados a problemática desta dissertação e a solução adotada para a sua resolução, incluindo a arquitetura funcional do sistema desenvolvido e a explicação dos princípios associados à sua elaboração.

No capítulo 4 são apresentados a modelização da solução e os conceitos teóricos fundamentais para a transformação do referencial 2D da câmara para o referencial 3D do mundo.

No capítulo 5 é apresentado o trabalho efetuado na aquisição dos dados de visão utilizados nesta dissertação, assim como dados posicionais do módulo GPS, e apresentado um pequeno protótipo de uma estação de aquisição, meramente como prova de conceito.

No capítulo 6 é apresentado o sub-sistema responsável pelo processamento dos dados de visão adquiridos pelo módulo de aquisição. Tem descritos os vários sub-sistemas integrantes e explicação do seu funcionamento.

No capítulo 7 são presentes as conclusões retiradas da elaboração da presente dissertação e incluída uma análise de potenciais trabalhos futuros no âmbito desta dissertação.

Por último, estão presentes os anexos e referências bibliográficas.



## Capítulo 2

# Estado da Arte

Atualmente, os sistemas de vídeo-vigilância têm sido extensamente utilizados para monitorizar e seguir veículos, tanto em terra ([10], [11]), como no mar ([12], [13]), e em ambientes aeroespaciais [14], assim como pessoas em ambientes dinâmicos ([15], [16]). No entanto, o ambiente marítimo representa um desafio para a vídeo-vigilância automática devido à complexidade do ambiente observado; alterações na luminosidade graduais ou súbitas (por exemplo nuvens), alterações no movimento (por exemplo movimentos intencionais ou não-intencionais da câmara), objetos de fundo com alta frequência (por exemplo, ondas e chuva) e reflexos são alguns dos exemplos que contribuem para este desafio.

### 2.1 Veículos não tripulados

O destaque da aplicabilidade de veículos não tripulados é evidenciado na sua utilização em ambientes hostis ou de reduzida acessibilidade. Dado não ser necessária a presença humana a bordo, as dimensões dos veículos poderão ser reduzidas, a sua eficácia energética é, desde logo, melhorada e a carga possível de transportar poderá ser também aumentada.

Atualmente, podemos encontrar vários exemplos destas aplicações em situações reais e concretas, desde explorações espaciais de curto ou longo alcance [17], explorações marítimas profundas e ultra-profundas [18], desarme de minas ou dispositivos explosivos [19], entre muitos outros. Em comum, estas situações apresentam um perigo muito elevado para a atuação por parte de um humano. Através da utilização de veículos não tripulados, o risco passa a ser inexistente.

Podemos dividir os veículos não tripulados em 2 grupos distintos: pilotados remotamente, onde o controlo do veículo é conseguido por meio de um piloto numa estação remota, ou autónomos, no qual o veículo é capaz de observar o meio envolvente e navegar autonomamente. Nesta dissertação, discutir-se-á sobre os veículos autónomos.

Como tal, a diversidade de aplicações possíveis para veículos autónomos levou à criação de vários tipos de veículos, cada qual pré-concebido para atuações em certos tipos de ambientes. Temos, então, os *UAV's* para atuarem no ar, *UGV's* para atuarem em terra, *USV's* para atuações à superfície em meios aquáticos, *UUV's* utilizados em ambientes sub-aquáticos e veículos espaciais

não tripulados para missões espaciais. Neste documento, serão focados os *USV's*, designados no restante documento como embarcações autônomas.

### 2.1.1 Embarcações Autônomas

Existem atualmente diversas embarcações e veículos não tripulados para aplicações em ambientes aquáticos. Utilizados, principalmente, em operações de vigilância e reconhecimento [20], são também utilizados em missões de combate litoral, detecção de minas sub-aquáticas, investigação de objetos suspeitos, defesa anti-submarino, operações fluviais, segurança portuária, entre outros [21]. Com o crescente desenvolvimento, surgem oportunidades de aplicações no âmbito civil, em inspeção, montagem e reparação de linhas sub-aquáticas [22], manutenção e reparação de cascos de navios [23], análise química das águas [24], busca e salvamento, oceanografia, entre muitos mais.

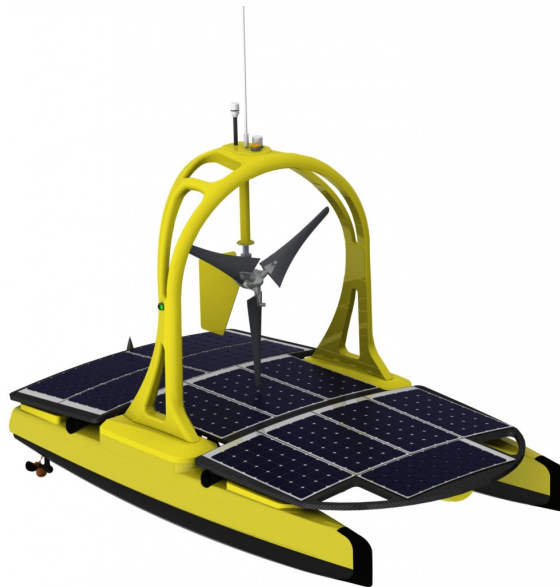


Figura 2.1: Exemplo de um USV com casco de catamarã (figura retirada de [1])

Como qualquer outro tipo de veículos, estas embarcações podem ser de pequeno, médio ou grande porte, sendo os 2 últimos, frequentemente, reestruturações de embarcações pré-existentes onde as preocupações com acomodação de tripulação são substituídas por acomodação de equipamento essencial à função da embarcação (por exemplo, embarcações semi-rígidas e catamarãs).

Um exemplo de uma embarcação autônoma, desenvolvida com um objetivo bem definido, é o veleiro autônomo da Faculdade de Engenharia da Universidade do Porto. Esta embarcação foi desenvolvida com o intuito de participar em competições de embarcações autônomas, sendo o seu sistema de controlo inteiramente autônomo e independente.

Com uma finalidade no âmbito mais profissional, existem outras embarcações pertinentes, como o *C-Worker* (figura 2.3), desenvolvido pela entidade *ASV Ltd.* para operações na indústria





Figura 2.2: FAST, Veleiro Autónomo da FEUP (figura retirada de [2])

do gás e petróleo; é capaz de posicionamento, inspeção e monitorização do meio circundante de uma forma totalmente autónoma [25]. Também pertinente, a embarcação *C-Enduro* é uma embarcação autónoma desenvolvida para recolher dados em alto mar na figura 2.4.

## 2.2 Sistemas de Visão

A estimação da posição no mundo de uma embarcação autónoma pode ser obtida através de sistemas de visão. Estes sistemas, nomeadamente de vídeo-vigilância, são capazes de detetar



Figura 2.3: Embarcação autónoma C-Worker para operações marítimas (figura retirada de [3])



Figura 2.4: Embarcação autónoma C-Enduro para recolher dados em alto mar (figura retirada de [4])

e monitorizar embarcações em ambientes aquáticos, servindo maioritariamente para efeitos de vigilância costeira [26].

### 2.2.1 Tipos de Sistemas de Visão

Existem atualmente diversos sistemas de visão, permitindo vários tipos de aplicações e desenvolvimentos. Fala-se de um sistema de visão no contexto de um sistema capaz de interpretar

imagens em duas ou três dimensões de um objeto. São exemplos os sistemas de aquisição via câmaras digitais e os sistemas de radar [27]. Será considerado, nesta dissertação, o primeiro.

Uma câmara digital é um sistema capaz de codificar imagens no espectro de luz visível, infravermelho ou ultravioleta. Em todas as câmaras digitais encontra-se um sensor de imagem que será do tipo *CCD* (*charge-coupled device*) ou *CMOS* (*Complementary metal-oxide-semiconductor*).

Dada a polivalência no espectro de luz capaz de adquirir, os sistemas de visão podem ser implementados em diferentes situações. Para efeitos de vigilância ou televisão existem as câmaras digitais capazes de codificar imagens no espectro de luz visível; para aplicações noturnas são utilizados sistemas de visão infravermelho; para explorações espaciais recorre-se, por vezes, a sistemas de visão ultra-violeta [28].

No contexto desta dissertação, falar-se-á dos sistemas de visão capazes de registar imagens no espectro visível, como sendo as câmaras digitais conhecidas.

## 2.3 Técnicas de Detecção de Objetos

Embora seja possível desenvolver sistemas de deteção de objetos baseados apenas em sensores óticos [29], sensores adicionais, tais como infravermelhos, GPS, AIS e, principalmente, o radar são normalmente emparelhados com vista a reduzir drasticamente a taxa de falsos positivos.

### 2.3.1 Radar

No entanto, embora a tecnologia do radar esteja bem estudada e forneça resultados bastante precisos, interferências no sinal de radar devido a aglomerados (*clutter*) tornam a deteção mais difícil [30]. Estes aglomerados são originados pelo retorno do sinal de radar (eco) de objetos indesejáveis, como por exemplo o chão, o mar, efeitos meteorológicos, edifícios, pontes ou outras embarcações. Para mais, embarcações não-metálicas e de reduzida dimensão não são detetadas e acresce ainda o custo de implementação e utilização elevado de um radar.

### 2.3.2 Subtração de Fundo

Subtração de fundo é uma técnica comum e largamente utilizada para criar uma máscara de primeiro plano (nomeadamente, uma imagem binária contendo os pixels pertencentes ao objeto alvo de deteção) utilizando, para isso, câmaras estáticas ([31], [32]).

Como indica o nome, esta técnica calcula a máscara de primeiro plano (*foreground*) através da subtração de um modelo do fundo à imagem em análise. Este modelo do fundo contém a parte estática da cena observada ou, de um modo geral, tudo o que possa ser considerado plano de fundo, dadas as características da imagem. Na figura 2.5 é possível observar-se o funcionamento desta técnica.

Denota-se, imediatamente, a inadequação desta técnica para imagens dinâmicas, onde a câmara não esteja estática. Com cada movimento da câmara tem que ser calculado o modelo de fundo.

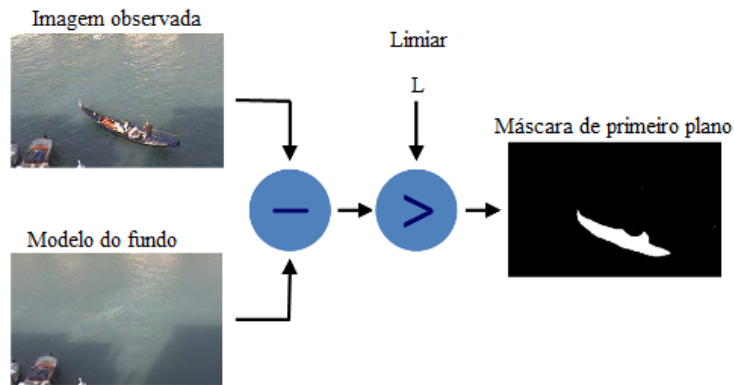


Figura 2.5: Diagrama da técnica de subtração de fundo.

### 2.3.3 Subtração de Fundo Adaptativa

Tornando mais robusta a subtração de fundo, a técnica de subtração de fundo adaptativa guarda, para cada pixel, um conjunto de valores adquiridos no passado para a mesma localização ou na vizinhança. De seguida compara este conjunto com o valor atual dos pixels, de forma a determinar se um pixel pertence ao plano de fundo, e adapta o modelo escolhendo aleatoriamente que valores substituir do modelo do fundo. Finalmente, quando um pixel é detetado como fazendo parte do plano de fundo, o seu valor é propagado a um pixel vizinho no modelo do fundo [33].

Um dos exemplos desta técnica é o algoritmo *ViBe*, cujo exemplo se apresenta na figura 2.6.



Figura 2.6: Exemplo da subtração de fundo adaptativa *ViBe*.

### 2.3.4 Classificadores

A técnica mais utilizada na deteção de objetos em cenas dinâmicas é, atualmente, a classificação baseada no *framework* de deteção de objetos proposto por Paul Viola e Michael Jones [34]. Este foi o primeiro *framework* a fornecer taxas de deteção de objetos em tempo real competitivas. Foi desenvolvido, principalmente, para deteção facial mas pode ser empregue na deteção de objetos variados. Através da computação de certas características da imagem analisada, é capaz de aferir com elevada taxa de sucesso se o objeto alvo se encontra na imagem.



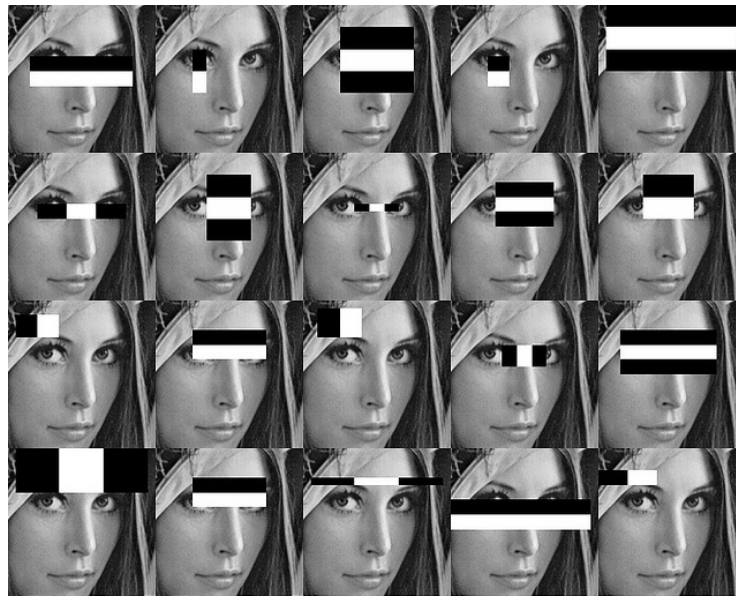


Figura 2.7: Exemplo da determinação de características Haar em várias imagens.

## 2.4 Trabalhos relacionados

Vários trabalhos foram já realizados no âmbito de deteção e monitorização de embarcações até à data, continuando a representar um desafio considerável. Enquanto que os sistemas, referidos no início deste capítulo, apresentem uma segmentação do primeiro plano robusta, não conseguem acomodar o ruído aglomerado com alta correlação espaço-temporal que a água apresenta. Em cenários aquáticos, as ondas causadas pelo vento ou o rasto de embarcações em movimento formam padrões de movimento com elevada correlação que afetam os sistemas tradicionais de subtração de fundo [35].

Existem diversos sistemas implementados que permitem a deteção de embarcações, por exemplo através da utilização de radares de abertura sintética (SAR) como descrito em [36]. Neste caso, é combinado o uso do satélite *TerraSAR-X* com o sistema AIS e o sistema de satélite AIS (*SatAIS*). No entanto, tratam-se de dois sistemas de difícil acesso, restringindo as possibilidades de sucesso.

Outros sistemas combinam sistemas de câmaras óticas e de infravermelhos (figura 2.9), radar e AIS. A deteção é efetuada estimando o movimento do plano de fundo e segmentando-o nos seus componentes. No entanto, deteções baseadas em movimento podem encontrar dificuldades quando uma embarcação se desloca diretamente na direção da câmara ou está ancorada ao largo da costa, devido à variação mínima entre fotogramas [37].

Um sistema visual automático para segurança marítima é também apresentado por Nuno Pires [38], no qual é utilizada uma câmara infravermelhos, GPS e AIS. Para detetar objetos relevantes, a área marítima é segmentada e a sua distribuição estatística é calculada; quaisquer irregularidades nesta distribuição é indicativo de um objeto de interesse. No entanto, devido a rastros de embarcações, tal abordagem pode levar a um crescente número de falsos positivos.

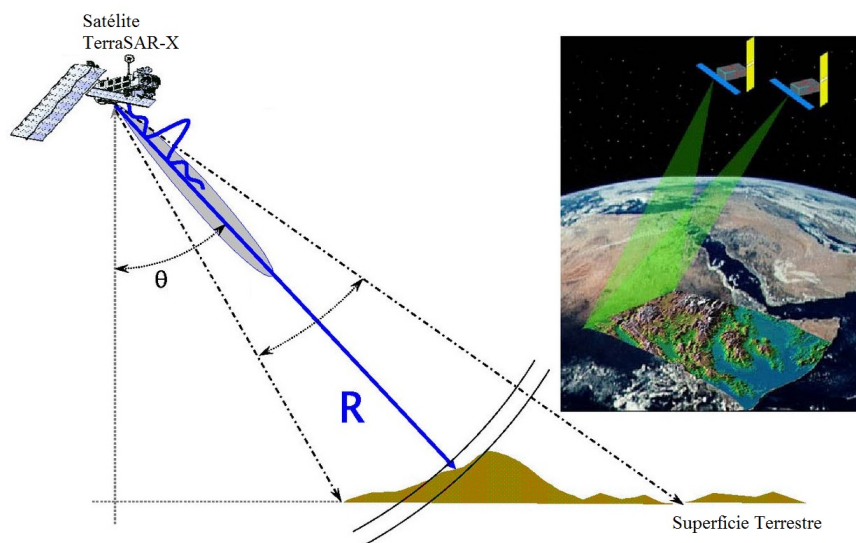


Figura 2.8: Ilustração do sistema de radar TerraSAR-X.

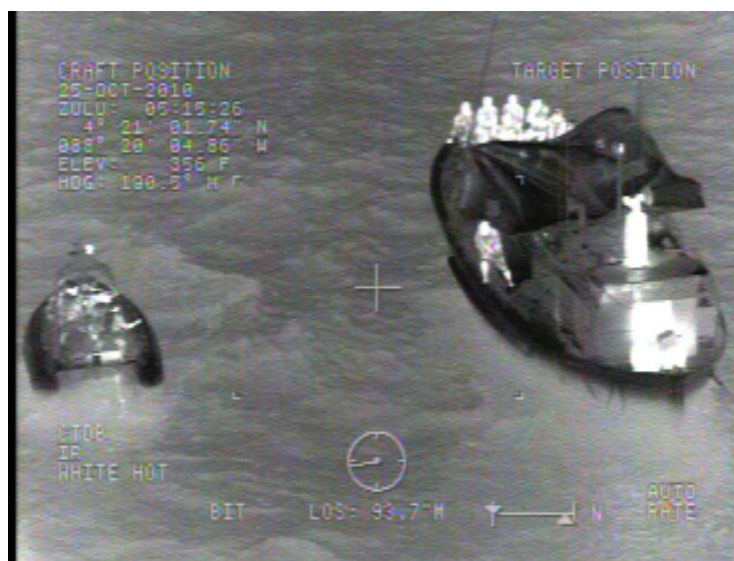


Figura 2.9: Aquisição de imagem por um sistema visual infravermelho (figura retirada de [5])

Uma outra abordagem à detecção de embarcações em vídeos de ambientes marítimos é apresentada por R. Wijnhoven [39]. Esta abordagem é baseada no Histograma de Gradientes Orientados [40]. Dado que a computação das características de detecção necessita de uma elevada capacidade computacional, o desempenho em tempo-real só é conseguido com recurso a aceleração do *hardware* com componentes programáveis como FPGAs.

Em conclusão, a análise da literatura demonstra uma série de críticas relacionadas com a utilização de câmaras PTZ, capazes de efetuar panorâmicas, assim como a presença de objetos alvo de dimensão variada, reflexos e rastros de embarcações na superfície da água e a presença de

embarcações aparentemente imóveis, ancoradas ao largo da costa ou em alto mar.

Esta dissertação visa abordar todas estas questões e os problemas encontrados.





## Capítulo 3

# Abordagem ao problema

Neste capítulo é apresentada a problemática da presente dissertação e a abordagem adotada para a sua resolução. Será descrita a formulação do problema, onde será explicada a origem desta problemática, seguida da solução proposta e de uma breve descrição.

### 3.1 Formulação do Problema

A navegação de embarcações autónomas é, regra geral, controlada, independentemente, pela própria embarcação, seguindo comandos estabelecidos pré-navegação ou em tempo real por um controlador humano e recorrendo a sensores de bordo para a análise do ambiente envolvente e consequente tomada de decisões de rota. Desta forma, cada embarcação é independente de outras embarcações, no sentido que desconhecem a mútua existência dentro do mesmo espaço de manobra assim como desconhecem a orientação dessas mesmas embarcações. Esta falta de informação pode ter consequências desastrosas se o ambiente de navegação estiver sobre-lotado e o espaço de manobra de cada embarcação for limitado. Por outro lado, estas embarcações são dependentes, unicamente, dos sensores de bordo, com os quais baseiam todas as suas decisões, colocando um problema relativamente ao desconhecimento da validade das informações dos sensores; a aquisição dos dados dos sensores de bordo, por exemplo *GPS*, carecem de uma validação e são aceites como fiáveis pela embarcação.

Não se dispondo de uma forma de comparar as leituras dos sensores de bordo com dados obtidos por terceiros foi encontrado aqui um problema que necessitava ser solucionado. Surge a necessidade de monitorizar e controlar estas embarcações, cuja navegação toma lugar em ambiente não exclusivo, de forma a tornar a sua navegação mais robusta e de forma a validar os dados posicionais da mesma, podendo ser criado um registo das posições de todas as embarcações para implementação de um controlador de tráfego autónomo.

Esta dissertação apresenta, então, uma solução para o problema em questão.

## 3.2 Solução Proposta

A solução proposta é a utilização de um sistema de visão monocular autónomo, divisível em 2 subsistemas - estação de aquisição e estação de processamento - capaz de detetar, monitorizar e acompanhar as movimentações das embarcações autónomas, criando um registo de toda a atividade costeira, acessível às embarcações, podendo estas efetuar uma validação aos seus dados posicionais e estabelecer a melhor rota com as informações de tráfego disponibilizadas. A solução deste problema resolve também o problema de vigilância costeira, pois o sistema desenvolvido é polivalente no que toca ao tipo de embarcações detetadas. Este sistema poderá ser instalado em qualquer ponto da zona costeira; várias estações de aquisição são instaladas ao longo da costa e são responsáveis por adquirir e transmitir os dados visuais para uma estação de processamento central que efetua uma análise destes dados, e cria um registo posicional das embarcações detetadas.

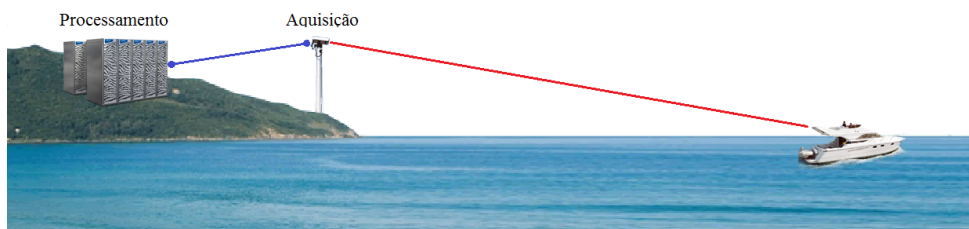


Figura 3.1: Ilustração da solução proposta.

### 3.2.1 Arquitectura funcional da solução

É, então, necessário compreender os pontos chave do modelo proposto e identificar os sistemas integrantes.

Conforme se pode observar pela figura 3.2, o sistema global é dividido em 2 sub-sistemas: estação de aquisição e estação de processamento.

### 3.2.2 Descrição do Sistema

Após instalação da estação de aquisição num ponto de vantagem de observação, esta é responsável por recolher e enviar os dados de visão adquiridos pela câmara e as informações dos dispositivos *IMU* e *GPS* para a estação de processamento central, através de uma rede de área local. De modo a oferecer a possibilidade de portabilidade da estação de aquisição, o módulo *GPS* permitirá obter as coordenadas da estação de aquisição no mundo, assim como a altitude face ao nível médio do mar. Através do sensor *IMU* é possível obter a orientação da câmara, relativamente aos 3 graus de liberdade de rotação, para estimação das coordenadas das embarcações detetadas no mundo.

Por sua vez, a estação de processamento interpreta os dados recebidos. Primeiramente, executa o processamento dos dados de visão recolhidos pela câmara; este processamento consiste na

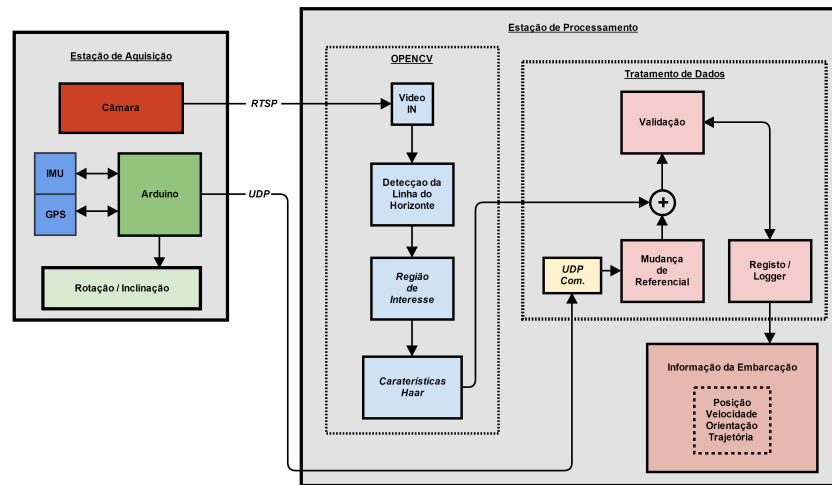


Figura 3.2: *Arquitectura funcional da solução proposta*

deteção da linha do horizonte, na imagem recebida, seguida da segmentação da região de interesse e aplicação do algoritmo de deteção de embarcações baseado num classificador em cascata com características *Haar*; de seguida efetua uma validação das deteções efetuadas e recebe e interpreta as informações dos dispositivos *IMU* e *GPS* fornecidas pela estação de aquisição; ultimamente, somando os resultados da anterior validação e da interpretação dos dados posicionais, faz o registo das embarcações detetadas com a sua posição no mundo, velocidade e orientação.

### 3.3 Síntese

O sistema desenvolvido partiu da necessidade de se monitorizar e controlar o tráfego marítimo autónomo de forma a poder ser implementado, no futuro, um ambiente estável e robusto que permita navegação de embarcações autónomas em espaços de navegação não exclusivos e que permita que esta seja segura e fiável.

Deste modo, o desenvolvimento do sistema adotado apoia-se em vários critérios de escolha, positivos e negativos. A implementação deste sistema permite a existência de várias estações de aquisição, instaladas em pontos de observação diferentes, ligadas a uma estação de processamento central; esta centralização permite maior capacidade de processamento, uma vez que pode ser dimensionada de forma a poder gerir os dados de várias estações de aquisição, e torna a manutenção e a identificação e correção de erros mais fácil. A modularidade desta solução oferece algum nível de redundância na aquisição de imagem, pois em caso de avaria ou defeito de uma estação de aquisição, outra estação na vizinhança pode ficar encarregue de observar a zona da estação defeituosa. No entanto, a transmissão dos dados de imagem é alvo de atrasos temporais quando comparada com um processamento local; também todo o processamento dos dados de visão se apoia numa única estação de processamento que, em caso de falha, condiciona todo o sistema.

Contudo, devido a restrições no acesso a embarcações autónomas para uma implementação correta desta abordagem, foi considerado que seriam alvo de deteção todo o tipo de embarcações, de forma a permitir testar o programa desenvolvido. Caso contrário, seria muito difícil obter cenários com embarcações autónomas, tanto para desenvolvimento do detetor de embarcações como para efeitos de teste e validação. No entanto, é seguro afirmar que este programa seria facilmente adaptado para incluir embarcações autónomas nos critérios de deteção.

Em suma, a solução desenvolvida foi ponderada após um balanço de todos os pontos positivos e negativos e foi escolhida como a de mais fácil implementação e sem exigência de recursos mais exigentes, como a utilização de radar ou *VTS*.

## Capítulo 4

# Modelização da Solução Proposta

### 4.1 Geometria Projetiva

A geometria projetiva serve como um *framework* para multi-visão 3D e para computação gráfica 3D. É utilizada para modelizar o processo de formação de imagem e gerar imagens sintéticas, ou reconstruir objetos 3D a partir de múltiplas imagens. Para modelizar retas, planos ou pontos num espaço 3D é utilizada normalmente a geometria Euclidiana. No entanto, a desvantagem da geometria Euclidiana é o facto de pontos no infinito não podem ser modelados e são considerados um caso especial. Este caso especial pode muito bem ser ilustrado através de um desenho em perspetiva de duas retas. Em perspetiva, duas retas paralelas, como por exemplo os carris de um caminho de ferro, encontram-se no infinito no ponto de fuga (figura 4.1). No entanto, a interseção das retas paralelas no infinito não é facilmente modelado pela geometria Euclidiana. Uma segunda desvantagem da geometria Euclidiana é que projetar um ponto 3D num plano de imagem requer uma operação de escalamento da perspetiva. Como o fator de escalamento é um parâmetro, a operação de escalamento da perspetiva exige uma divisão que se torna uma operação não-linear. Assim, decidiu-se evitar o uso da geometria Euclidiana.



Figura 4.1: *Dois retas paralelas interseccionam-se no infinito, no ponto de fuga (figura retirada de [6]).*

A geometria projetiva é, então, um *framework* bastante atraente para contornar as desvantagens supra descritas da geometria Euclidiana. Num espaço Euclidiano, um ponto definido em três dimensões é representado por um vetor de três elementos  $(X, Y, Z)^T$ ; no espaço projetivo, este mesmo ponto é definido através de um vetor de 4 elementos  $(X_1, X_2, X_3, X_4)^T$  tal que:

$$X = \frac{X_1}{X_4}, Y = \frac{X_2}{X_4}, Z = \frac{X_3}{X_4}, \quad (4.1)$$

onde  $X_4 \neq 0$ . Regra geral, as coordenadas  $(X, Y, Z)^T$  e  $(X_1, X_2, X_3, X_4)^T$  são designadas coordenadas não homogêneas e coordenadas homogêneas, respetivamente.

Generalizando, o mapeamento a partir de um ponto no espaço Euclidiano  $n$ -dimensional para um espaço projetivo  $(n+1)$ -dimensional pode ser escrito como:

$$\overbrace{(X_1, X_2, \dots, X_n)^T}^{\text{Espaço Euclidiano}} \rightarrow \overbrace{(\lambda X_1, \lambda X_2, \dots, \lambda X_n)^T}^{\text{Espaço Projetivo}}, \quad (4.2)$$

onde  $\lambda \neq 0$  corresponde a um parâmetro de escalamento livre. Este parâmetro é normalmente denominado fator de escalamento homogêneo.

Utilizando o *framework* de geometria projetiva apresentado, é apresentado, de seguida, o modelo da câmara *pin-hole*.

## 4.2 Modelo de uma Câmara *pin-hole*

Nesta secção é descrito o processo de aquisição de imagem, conhecido como o modelo da câmara *pin-hole*, o qual é empregue nesta dissertação. Mais especificamente, será discutido o modelo que integra os parâmetros extrínsecos e intrínsecos da câmara, tais como a distância focal e a distorção da lente; seguidamente, é estendido o modelo da câmara simples para integrar parâmetros extrínsecos e intrínsecos da câmara correspondendo à posição e orientação da câmara.

### 4.2.1 Parâmetros Intrínsecos da Câmara

O modelo da câmara *pin-hole* define a relação geométrica entre um ponto 3D e a sua projeção 2D correspondente no plano da imagem. Quando o modelo da câmara *pin-hole* é utilizado, este mapeamento geométrico, de 3D para 2D, é chamado de projeção em perspetiva. Designa-se o centro da projeção em perspetiva (o ponto no qual todas as retas se intersectam) como o centro ótico ou o centro da câmara e a linha perpendicular ao plano da imagem e passando pelo centro ótico como o eixo ótico (ver figura 4.2). Adicionalmente, o ponto de interseção do plano da imagem com o eixo ótico é chamado de ponto principal. O modelo de projeção em perspetiva de uma câmara *pin-hole* é, então, descrito de seguida.

### 4.2.1.1 Projeção em Perspectiva através de coordenadas homogêneas

Considere-se uma câmara com o eixo ótico colinear com o eixo  $Z_{cam}$  e o centro ótico localizado na origem de um referencial 3D (figura 4.2).

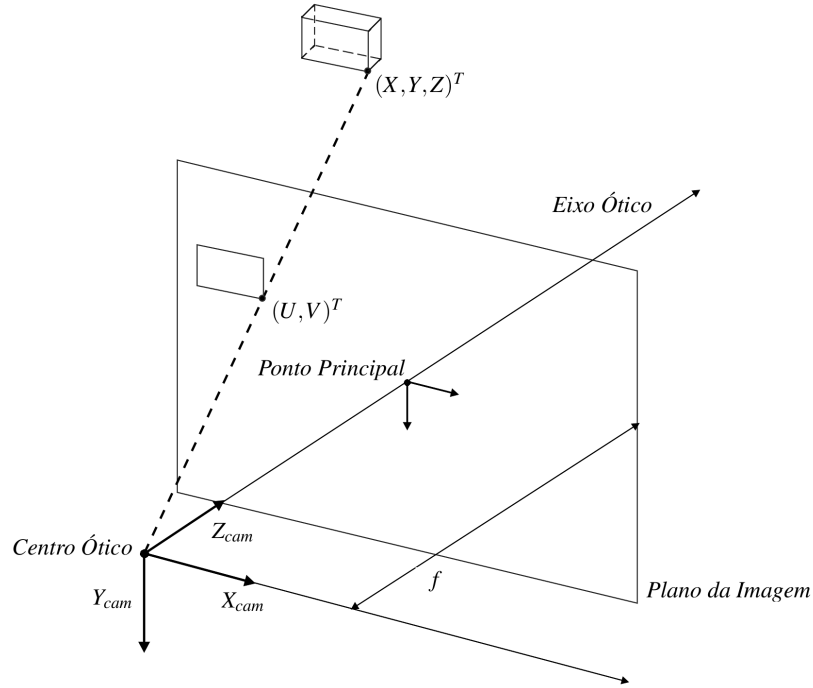


Figura 4.2: O modelo ideal de uma câmara pin-hole descreve a relação entre um ponto 3D  $(X, Y, Z)^T$  e a sua respetiva projeção 2D  $(U, V)^T$  no plano da imagem.

A projeção de um ponto 3D do mundo  $(X, Y, Z)^T$  no plano da imagem no pixel com localização  $(u, v)^T$  pode ser escrita como:

$$u = \frac{Xf}{Z}$$

e

$$v = \frac{Yf}{Z}, \quad (4.3)$$

onde  $f$  é a distância focal. Para evitar tal operação de divisão não linear, a equação anterior pode ser reformulada através do *framework* da geometria projetiva, como:

$$(\lambda u, \lambda v, \lambda)^T = (Xf, Yf, Z)^T. \quad (4.4)$$

Esta fórmula pode ser descrita na forma matricial como:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (4.5)$$

onde  $\lambda = Z$  é o fator de escalamento homogéneo.

#### 4.2.1.2 Offset do Ponto Principal

A maior parte dos sistemas de processamento de imagem definem a origem do referencial dos pixels no canto esquerdo-superior da imagem. No entanto, foi previamente assumido que a origem do referencial dos pixels corresponde ao ponto principal  $(o_x, o_y)^T$ , localizado no centro da imagem (figura 4.3). A conversão do referencial é, desta forma, necessária; através de coordenadas homogéneas, a posição do ponto principal pode ser rapidamente integrada numa matriz de projecção. A equação de projecção em perspetiva fica, desta forma:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & o_x & 0 \\ 0 & f & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (4.6)$$

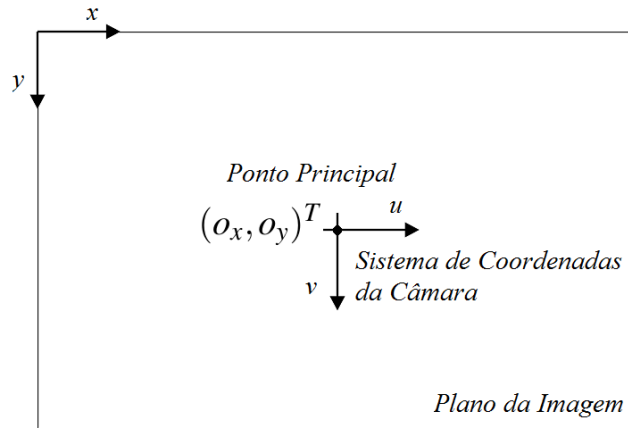


Figura 4.3: Os referenciais da imagem  $(x,y)$  e da câmara  $(u,v)$ .

#### 4.2.1.3 Características do sensor de imagem

Para derivar a fórmula descrita na equação 4.6, foi implicitamente assumido que os pixels do sensor de imagem são quadrados, isto é, a proporção da tela é 1:1 e os pixels não estão desviados (*skewed*). No entanto, ambas as suposições poderão nem sempre ser válidas. Primeiro, por



exemplo, um sistema NTSC define pixels não-quadrados com uma proporção de tela de 10:11 (na prática, a proporção de tela dos pixels é frequentemente fornecida pelo produtor do sensor de imagem); segundo os pixels podem potencialmente estar desviados, especialmente no caso da imagem ser adquirida por um digitalizador de imagem. Neste caso em particular, a grelha de pixels pode estar desviada devido a uma sincronização imprecisa no processo de amostragem dos pixels. Ambas as imperfeições do sistema de imagem mencionadas anteriormente podem ser levadas em consideração pelo modelo da câmara, utilizando para isso os parâmetros  $\eta$  e  $\tau$ , os quais modelizam a proporção de tela e desvio dos pixels, respetivamente (figura 4.4).

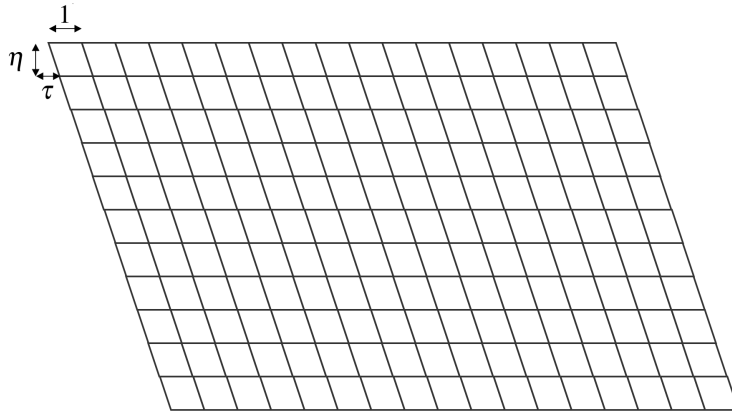


Figura 4.4: Sensor de imagem não ideal com pixels desviados e não quadrados.

O mapeamento da projeção pode, agora, ser atualizado como:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & \tau & o_x & 0 \\ 0 & \eta f & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} K & 0_3 \end{bmatrix} P, \quad (4.7)$$

sendo  $P = (X, Y, Z, 1)^T$  um ponto 3D definido com coordenadas homogêneas. Na prática, utilizando câmaras digitais recentes, pode ser seguramente assumido que os pixels são quadrados ( $\eta = 1$ ) e não sofrem desvio ( $\tau = 1$ ). A matriz de projeção que incorpora os parâmetros intrínsecos é designada como  $K$  e o vetor de zeros designado  $0_3$ .

#### 4.2.1.4 Distorção Radial da Lente

Na realidade, as lentes das câmaras sofrem tipicamente de distorção não-linear. Na prática, a distorção radial das lentes faz com que linhas retas sejam mapeadas como linhas curvas. Como se observa na figura 4.5, a distorção radial da lente aparece mais visível nos bordos da imagem, onde a distância radial é alta. Uma técnica padrão para modelizar a lente radial é descrita em seguida.

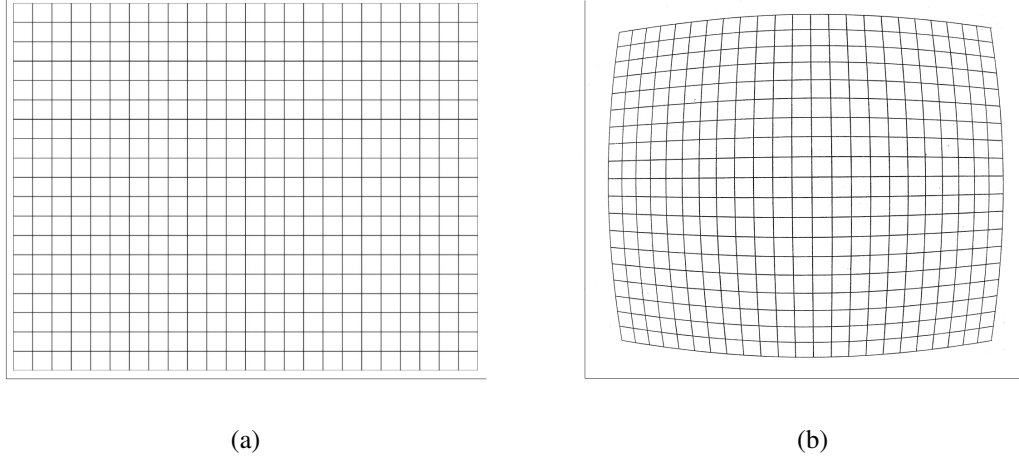


Figura 4.5: As lentes das câmaras sofrem distorção radial. Grelha de pixels de uma imagem não distorcida 4.5a e distorcida 4.5b

Sejam  $(x_u, y_u)^T$  e  $(x_d, y_d^T)$  as posições dos pixels corretos e distorcidos, respetivamente, tem-se que a relação entre um pixel distorcido e um pixel não distorcido pode ser modelizada com uma função polinomial e escrita como:

$$\lambda \begin{pmatrix} x_u - o_x \\ y_u - o_y \end{pmatrix} = L(r_d) \begin{pmatrix} x_d - o_x \\ y_d - o_y \end{pmatrix}, \quad (4.8)$$

onde

$$L(r_d) = 1 + k_1 r_d^2 \text{ e } r_d^2 = (x_d - o_x)^2 + (y_d - o_y)^2 \quad (4.9)$$

sendo  $k_1$  é o coeficiente da distorção radial.

No caso de  $k_1 = 0$ ,  $x_u = x_d$  e  $y_u = y_d$ , o que corresponde à ausência de distorção radial.

De salientar que a partir da equação 4.8 obtém-se a posição do pixel correta em função da posição distorcida do pixel. Dado que não é do interesse desta dissertação apresentar a imagem não distorcida, mas sim retirar as informações corrigidas da imagem, não será explorado o método de mapeamento inverso para obter a imagem não distorcida. No entanto, o método de correção da distorção da lente requer o conhecimento dos parâmetros da lente, isto é,  $k_1$  e  $(o_x, o_y)^T$ . A estimação destes parâmetros é calculada minimizando uma função de custo que mede a curvatura das linhas na imagem distorcida. Esta medição é conseguida através da deteção de pontos característicos pertencentes à mesma linha num aparelho de calibração, por exemplo, um padrão em xadrez para calibração [41]. Cada ponto pertencente à mesma linha, na imagem distorcida, forma uma linha curva em vez de uma linha reta [42]. Comparando o desvio entre a linha curva e o modelo teórico da linha reta, os parâmetros de distorção são calculados.

### 4.2.2 Parâmetros Extrínsecos da Câmara

Em oposição aos parâmetros intrínsecos que descrevem os parâmetros internos da câmara (distância focal, parâmetros da lente radial), os parâmetros extrínsecos indicam a posição e orientação externas da câmara no mundo 3D. Matematicamente, a posição e orientação da câmara é definida por um vetor  $C$  e por uma matriz de rotação  $R$  (figura 4.6).

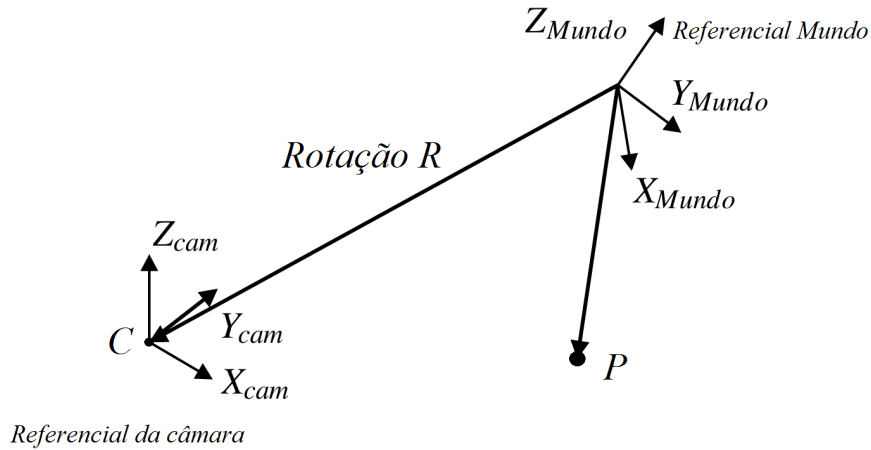


Figura 4.6: A relação entre o referencial da câmara e o do Mundo é definida pelo centro da câmara  $C$  e pela rotação  $R$  da câmara.

Para obter a posição de um pixel  $p = (x, y, 1)^T$  de um ponto  $P$  homogêneo no Mundo, a câmara sofre primeiro uma translação para a origem do referencial mundo e de seguida sofre uma rotação. Isto é descrito pela equação 4.10:

$$\lambda p = \begin{pmatrix} K | 0_3 \end{pmatrix} \begin{bmatrix} R & 0 \\ 0_3 & 1 \end{bmatrix} \begin{bmatrix} 0_3^T & -C \\ 0 & 1 \end{bmatrix} P. \quad (4.10)$$

Alternativamente, combinando as matrizes, a equação 4.10 pode ser reformulada como:

$$\lambda p = \begin{bmatrix} K | 0_3 \end{bmatrix} \begin{bmatrix} R & -RC \\ 0_3^T & 1 \end{bmatrix} P = KR \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - KRC. \quad (4.11)$$

### 4.2.3 Projeção de um ponto 2D em 3D

Na subsecção anterior, foi descrito o processo de projetar um ponto 3D no plano 2D da imagem. Segue-se a descrição de como um ponto 2D pode ser inversamente projetado no espaço 3D e calcular as respetivas coordenadas no referencial do mundo.

Considerando um ponto 2D numa imagem, existe uma coleção de pontos que são mapeados e projetados no mesmo ponto  $p$ . Este conjunto de pontos 3D constitui uma reta que liga o ponto

$C = (C_x, C_y, C_z)^T$  e  $p = (x, y, 1)^T$ . Da equação 4.13, a reta  $P(\lambda)$  associada ao pixel  $p = (x, y, 1)^T$  pode ser definida como:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \overbrace{C + \lambda R^{-1} K^{-1} p}^{\text{reta } P(\lambda)}, \quad (4.12)$$

onde  $\lambda$  é o fator de escalamento positivo definindo a posição do ponto 3D na reta  $P(\lambda)$ .

A coordenada  $Z$  é facilmente obtida da altura da maré no local específico. Consideremos um *offset*,  $Z_{offset}$ , como sendo a diferença entre a altura do nível médio do mar, no qual é baseada a altura devolvida pelo GPS, e a altura da maré naquele momento; desta forma,  $Z = Z_{offset}$ . Conhecendo  $Z$ , é possível obter as coordenadas  $X$  e  $Y$  calculando  $\lambda$  através da fórmula:

$$\lambda = \frac{Z - C_z}{z_3}, \text{ onde } (z_1, z_2, z_3)^T = R^{-1} K^{-1} p. \quad (4.13)$$

#### 4.2.4 Calibração da Câmara

A calibração da câmara envolve a estimação de ambos os parâmetros extrínsecos e intrínsecos. Para o cálculo dos parâmetros da câmara, é utilizada uma técnica prática baseada num equipamento de calibração [43]. Para a correção da distorção da lente, a estimação dos parâmetros da câmara é baseada num equipamento de calibração com geometria conhecida, tal como um tabuleiro de xadrez. Usando várias vistas de perspetiva do tabuleiro, o algoritmo, com conhecimento da geometria do tabuleiro, estima a posição, orientação e parâmetros internos da câmara. Este processo de calibração é designado calibração forte.

Para obter os parâmetros extrínsecos da câmara, são utilizadas as informações dos módulos GPS e IMU. As coordenadas da câmara serão  $(C_x, C_y, C_z)$  onde  $C_x = C_y = 0$  e  $C_z = \text{altitude}$  medida através do módulo GPS.

O módulo IMU está rigidamente ligado à câmara. Através da informação da orientação do IMU, podemos obter a matriz de rotação  $R$ . Isto é conseguido através de métodos projetados para calibração do conjunto da câmara e IMU ([44],[45], [46], [47])

Com recurso ao módulo GPS podem determinar-se as coordenadas geográficas da câmara. Desta forma, as coordenadas do ponto  $(X, Y, Z)$ , em termos de longitude e latitude podem ser facilmente obtidas através de cálculos simples. Sabendo a latitude da localização da câmara, é possível correlacionar distâncias (por exemplo em metros) com as coordenadas geográficas do módulo GPS, através de uma relação, por exemplo, entre a distância em metros de um grau de latitude e um grau de longitude; devido à forma elíptica da Terra, o primeiro passo será obter a latitude da estação de aquisição para seguidamente se correlacionar graus de longitude e latitude em metros. Admitindo os eixos do referencial da câmara alinhados com Norte e Oeste, a longitude e latitude da embarcação detetada será dada pelas coordenadas da câmara mais a projeção nos eixos  $XY$  após correlação das medidas anteriores, como se pode observar pela figura 4.7.

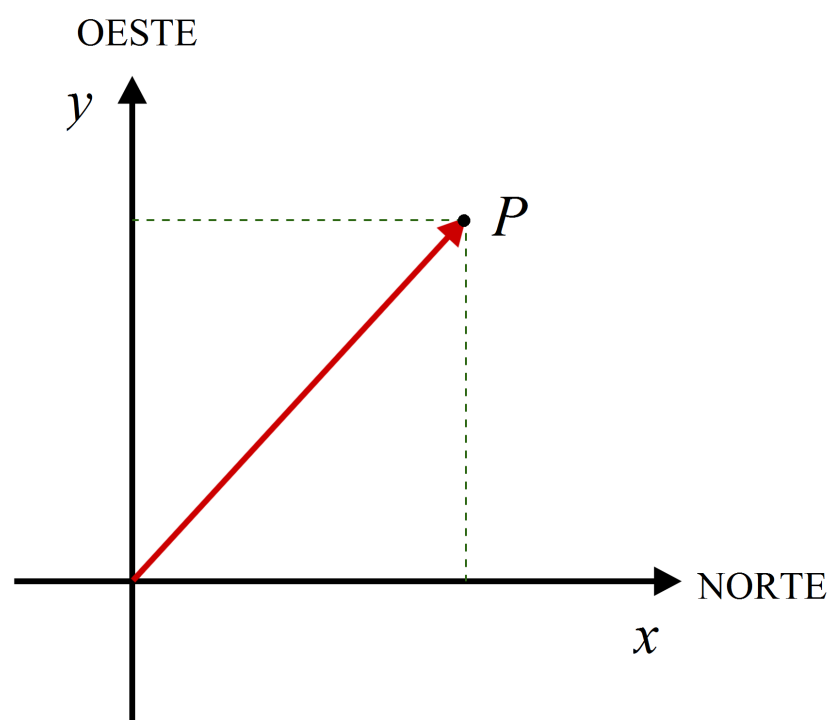


Figura 4.7: Estimação das coordenadas de uma embarcação no ponto  $P$ , através da projeção das suas coordenadas nos eixos alinhados com Norte e Oeste.

Desta forma, obtêm-se todos os parâmetros necessários para calcular a posição de qualquer ponto no espaço 3D, com recurso às fórmulas apresentadas anteriormente.

### 4.3 Aquisição

A aquisição dos dados de visão e dados posicionais da câmara é feita na estação de aquisição.

Os dados de visão são adquiridos com recurso a uma câmara PTZ (figura 4.8), de forma a permitir efetuar panorâmicas laterais e verticais das zonas de observação. De forma a minimizar os atrasos temporais na transmissão dos dados de visão, a câmara comunicará com a estação de processamento através de uma rede de área local e utilizando o protocolo RTSP; este protocolo permite que o cliente (neste caso a estação de processamento) controle fluxos de multimédia em tempo real do servidor. RTSP permite que o cliente envie pedidos ao servidor controlando a reprodução de multimédia mas não trata da comunicação ele mesmo; esta é feita sob o protocolo TCP-IP.



Figura 4.8: Exemplo de uma câmara PTZ (figura retirada de [7]).

Os dados posicionais da estação de aquisição são adquiridos com recurso a um módulo GPS e um módulo IMU, montados na própria estação. Estes dados são transmitidos através da mesma rede de área local mas sob o protocolo UDP. Com os dados obtidos pelo módulo GPS, é possível determinar a posição da estação de aquisição no mundo, em termos de altura em relação ao nível médio do mar, coordenadas no referencial mundo e a etiqueta temporal. Através do módulo inercial, retiram-se os parâmetros extrínsecos da câmara, tais como as orientações nos 3 eixos de rotação, transversal, longitudinal e vertical (figura 4.9).

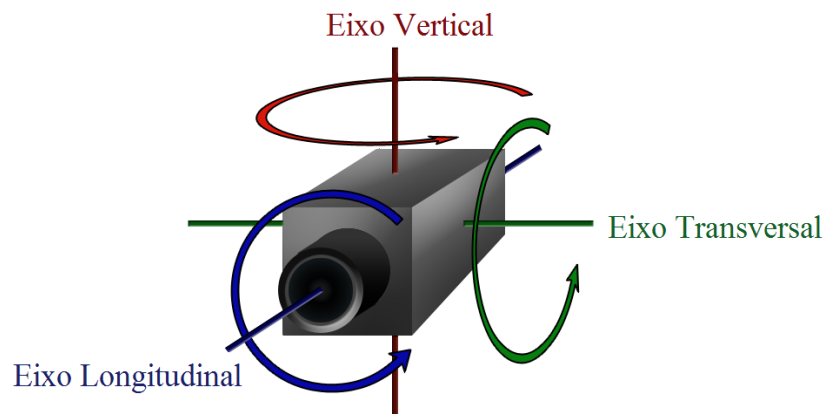


Figura 4.9: Eixos de rotação da câmara.

## 4.4 Processamento

O processamento dos dados recebidos da estação de aquisição é efetuado na estação de processamento. Esta estação é uma plataforma de recepção dos dados enviados pelas estações de aquisição e subsequente tratamento e interpretação dos mesmos para extração das informações relevantes relativas às embarcações detetadas.

### 4.4.1 Entrada de Vídeo ou *Stream*

A aquisição dos dados de visão para análise tem duas abordagens: via vídeo pré-gravado ou por aquisição em tempo-real via câmara da estação de aquisição; serve a primeira, maioritariamente, para testes e análise do software desenvolvido de forma a permitir avaliar a eficácia e a consistência dos algoritmos desenvolvidos e compreender de que forma cada alteração ao código destes afeta o resultado final pois tem-se que os objetos de teste (as imagens retiradas do vídeo em análise) serão iguais entre testes; a aquisição em tempo-real via câmara da estação de aquisição será para execuções em tempo-real e implementação em cenário real, sendo a aquisição de imagem realizada no momento imediatamente anterior ao seu processamento.

A aquisição de imagem é, então, efetuada dentro de um ciclo *while* infinito no qual é, inicialmente, aberto o ficheiro vídeo a ser processado ou é acedido ao endereço da câmara que envia fotogramas em tempo-real (*streaming*); segue-se um segundo ciclo *while*, o qual se inicia com a captura de um fotograma da sequência de vídeo ou do fluxo da câmara, retendo uma imagem de 8 bits e 3 canais (*RGB*), seguida da execução dos algoritmos de processamento do fotograma capturado. Este segundo ciclo é repetido até se extrair o último fotograma da sequência de vídeo, após o qual retorna ao primeiro ciclo *while* e repete todas as operações anteriores, retomando o vídeo desde o início; no caso de estar a ser efetuada a captura de fotogramas via *stream* da câmara de aquisição, esta só será interrompida com o término do programa.

A arquitetura funcional deste algoritmo pode ser observada na figura 4.10.

Após a aquisição do fotograma para processamento, este é redimensionado para uma dimensão pré-definida, menor que a original, de forma a reduzir a carga computacional necessária para um processamento eficaz.

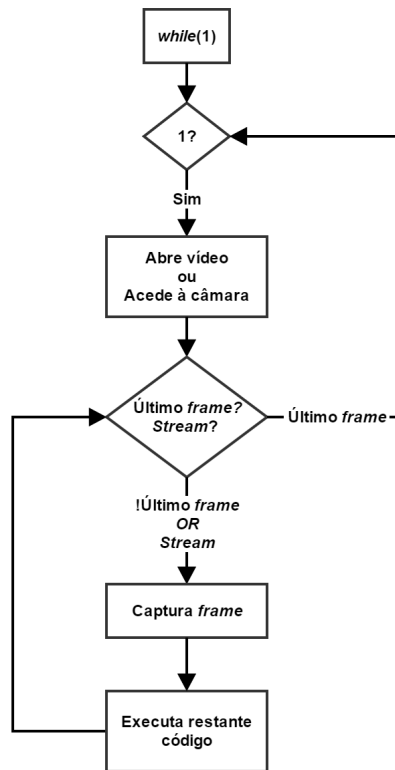


Figura 4.10: Diagrama do algoritmo de aquisição de imagem.

Após a aquisição da imagem e do seu redimensionamento, são aplicados vários algoritmos. De forma a reduzir a carga computacional imposta, é aplicado um algoritmo de deteção da linha do horizonte de forma a eliminar partes da imagens irrelevantes para o detetor de embarcações, isto é, descartar as áreas da imagem acima da linha do horizonte, onde não serão encontradas embarcações. Seguidamente é aplicado o detetor à imagem recortada e todas as deteções são registadas. Este registo é apenas um vetor de retângulos, no qual cada um é um indicador de uma embarcação detetada. Cada retângulo devolvido pelo detetor tem um ponto inicial associado, seguido da largura e altura do respetivo retângulo; desta forma, é possível saber-se em que ponto da imagem se detetou a embarcação (figura 4.11).

## 4.5 Pós-Processamento

Após a deteção positiva de embarcações pelo sub-sistema anterior procede-se então, ao pós-processamento dos dados de visão e ao cálculo da posição no mundo das embarcações. Para



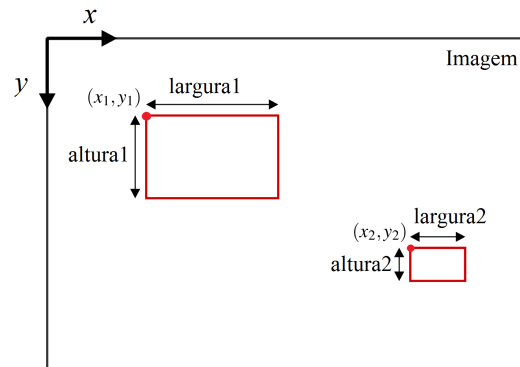


Figura 4.11: *Deteção de 2 "embarcações" numa imagem, como dois retângulos, e respectivas coordenadas e dimensões.*

tal, no sub-sistema de tratamento de dados, os objetos detetados pelo algoritmo de deteção são submetidos a uma análise e validação. Esta validação é feita recorrendo ao histórico de deteções anteriores; o histórico de deteções é implementado através de um mapa temporal adaptativo que evidencia as deteções ocorridas previamente com um grau de probabilidade de sucesso associado. Nesta etapa, o sub-sistema de mudança de referencial recebe os dados posicionais da estação de aquisição, via protocolo *UDP*, e com base nas informações recebidas calcula as posições das embarcações detetadas no referencial do mundo.

## 4.6 Registo e Tratamento da Informação

Uma vez validadas as deteções obtidas, é atualizado o registo que contém todas as embarcações detetadas até à altura, diferenciadas por um código identificador único, juntamente com as informações posicionais calculadas das embarcações. Este registo é o propósito máximo desta dissertação e, através deste modelo, é de consulta pública às entidades que pretendam informações de tráfego marítimo na zona supervisionada pelo sistema, sejam embarcações em navegação, autónomas ou não, assim como utilizadores terceiros que pretendam conhecer o ambiente de navegação alvo desta supervisão.

## 4.7 Biblioteca OpenCv

A biblioteca *OpenCV* (*Open Source Computer Vision*) é uma biblioteca de funções de programação preparada, maioritariamente, para visão computacional em tempo-real. É, como indica o nome, uma biblioteca de código aberto que permite a sua utilização e modificação pelo público geral, tendo sido, por isso, a escolha para o desenvolvimento desta dissertação.



## Capítulo 5

# Aquisição dos Dados de Visão e Posicionais

Neste capítulo é apresentado o trabalho efetuado na aquisição dos dados de visão e dos dados posicionais assim como é apresentado um pequeno protótipo de uma estação de aquisição, apenas para prova de conceito.

### 5.1 Aquisição de Vídeos

Com vista a obter cenários para teste do detetor de embarcações desenvolvido, foram adquiridos diversos vídeos, filmados na zona costeira no ponto mais alto encontrado. Para isso foi utilizada uma câmara *FullHD 1080*, ilustrada na figura 5.1.



Figura 5.1: *Câmara utilizada na aquisição dos vídeos de teste.*

Os vídeos foram gravados em duas condições climáticas diferentes, céu limpo e enublado, de forma a colocar um desafio maior na deteção da linha do horizonte.

Alguns exemplos de imagens captadas podem ser observados na figura 5.2.

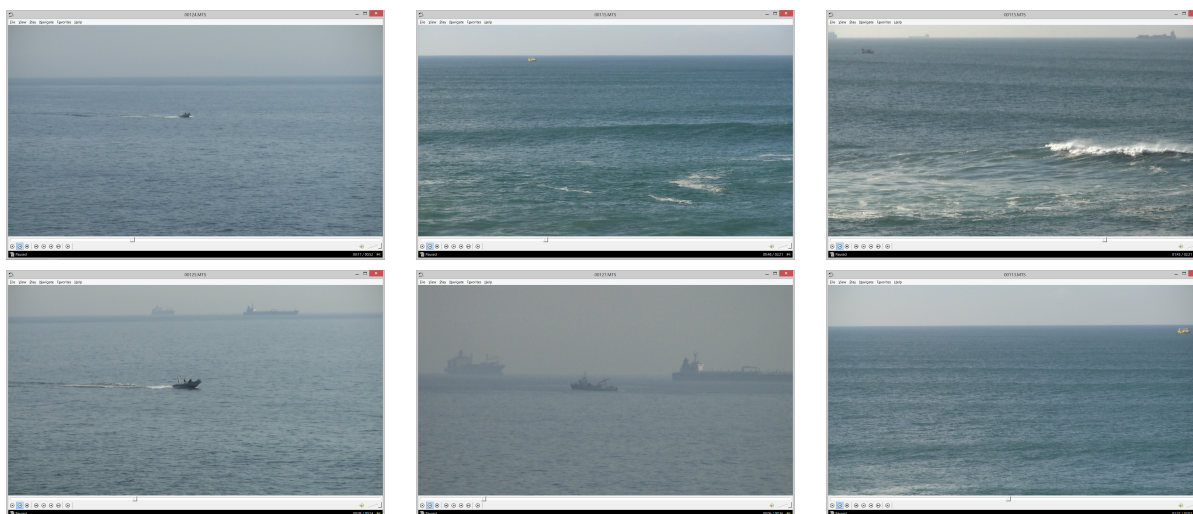


Figura 5.2: Exemplos de imagens captadas em diversos vídeos.

## 5.2 Estação de Aquisição

Como prova de conceito e também para testes em ambiente real, foi desenvolvido um pequeno protótipo de uma estação de aquisição composta por um módulo *Arduino*, um módulo GPS, uma câmara USB e um servo-motor (figura 5.3).

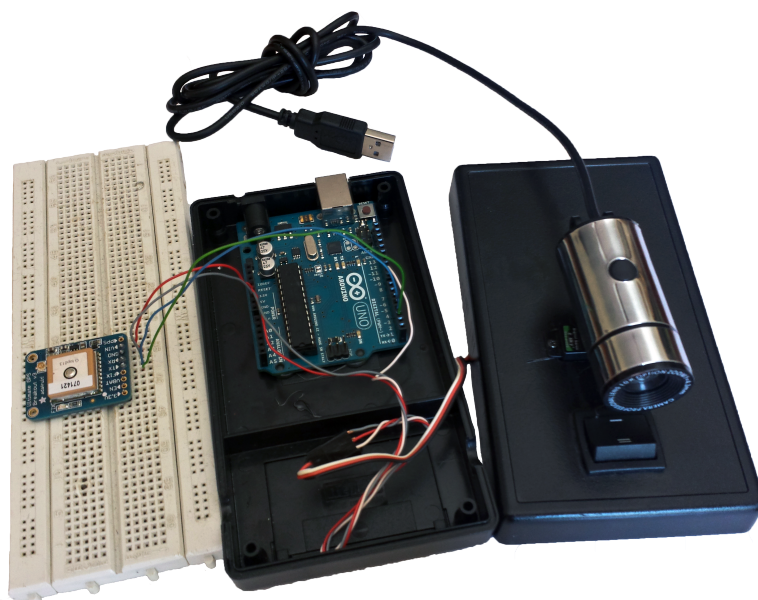


Figura 5.3: Protótipo de uma estação de aquisição.

Este protótipo visava aferir a qualidade de funcionamento do programa em tempo-real, intercalando o processamento de imagem com a leitura da porta-série e leitura dos dados do GPS, assim como a movimentação da câmara. Foi apenas utilizado para testar o programa desenvolvido em tempo-real, sendo que todos os testes dos algoritmos de visão foram efetuados sobre os vídeos capturados com a câmara descrita na secção anterior.

### 5.2.1 Módulo *Arduino* e Comunicação

Para comunicar com a estação de processamento, neste caso um computador pessoal, foi utilizado um *Arduino Uno* ligado por USB ao computador. Para comunicar com a estação e com os módulos a si ligados, foram adaptados algoritmos fornecidos pela entidade *Adafruit* [48] com vista a poder interagir com os módulos exteriores e comunicar com a estação de processamento.

Na estação de processamento é controlado o fluxo de informação com o *Arduino*, isto é, o *Arduino* fica à "escuta" da porta-série e apenas envia comunicações (no caso do GPS) ou efetua atuações (no caso do servo-motor) quando um pedido é efetuado do lado da estação, em funcionamento estilo *pooling*.

### 5.2.2 Aquisição do Sinal do GPS

O sinal GPS é obtido com recurso ao módulo *Adafruit Ultimate GPS Breakout* (figura 5.4). Após a instalação das bibliotecas *open source* próprias deste módulo, a extração do sinal torna-se banal. No *Arduino*, após interrupções de 1 em 1 milissegundos, é executada a rotina de interrupções ISR; nesta rotina o *Arduino* verifica se existem novos dados posicionais do GPS e, em caso afirmativo, regista as leituras do módulo de GPS através de uma ligação série por *software* ( $R_x, T_x$ ). Estas leituras são efetuadas de 10 em 10 segundos pelo módulo GPS. Seguidamente, o *Arduino* continua à "escuta" da porta-série, ligada à estação de processamento, até que lhe seja feito um pedido de envio das informações do sinal de GPS. No caso de não ser feito um pedido de envio da informação antes dos 10 segundos de intervalo de leitura do GPS, as informações anteriores são substituídas pelas mais recentes.

Na figura 5.5 observam-se algumas medições obtidas através deste sistema, ao mesmo tempo que o restante programa executava o detetor de embarcações (parte das coordenadas foram pixelizadas por razões de segurança).

### 5.2.3 Rotação do sistema de aquisição

Foi também implementado um sistema de rotação da câmara com recurso a um servo-motor digital, controlado por PWM (figura 5.6) por impossibilidade de obter uma câmara PTZ para testes. Mais uma vez, a implementação do controlador deste servo-motor no *Arduino* é relativamente simples recorrendo às bibliotecas próprias para o efeito.

Através de comandos na estação de processamento, é possível controlar se o servo-motor roda para a esquerda ou para a direita. Este módulo funcionou bastante bem, no entanto não é o mais indicado para esta situação específica pois possui um passo angular demasiado alto.

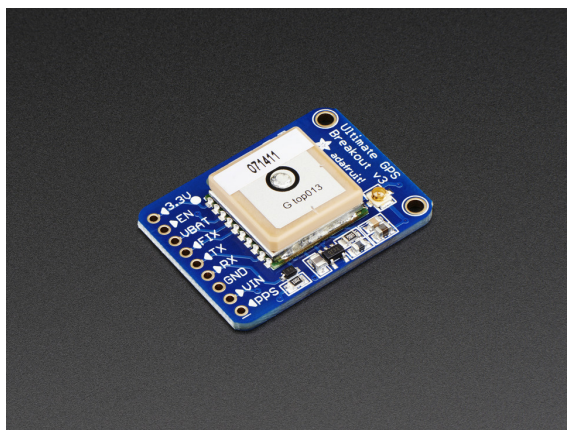


Figura 5.4: Módulo de GPS implementado na estação de aquisição (figura retirada de [8]).

```
C:\Users\Mauricio\Documents\Visual Studio 2013\Projects\Controlo Visual\D...
total_frames: 9048
--- Getting GPS ---
--->SEND g.
Asneira!
--- Getting GPS ---
--->SEND g.
incomingData: -8.6011;41.1911
longitude: -8.6011
latitude: 41.1911
--- Getting GPS ---
--->SEND g.
incomingData: -8.6011;41.1911
longitude: -8.6011
latitude: 41.1911
--- Getting GPS ---
--->SEND g.
incomingData: -8.6011;41.1911
longitude: -8.6011
latitude: 41.1911
--- Getting GPS ---
--->SEND g.
incomingData: -8.6011;41.1911
longitude: -8.6011
latitude: 41.1911
```

Figura 5.5: Módulo de GPS implementado na estação de aquisição.



Figura 5.6: Servo-motor utilizado na estação de aquisição para variação da pose da câmara (figura retirada de [9]).

### 5.3 Discussão de Resultados e Conclusão

Este protótipo serviu para testar se o conjunto estação de aquisição e estação de processamento funcionariam nas condições esperadas. É seguro, com base nos resultados apresentados na figura 5.5 e na figura 5.7 afirmar que a conjugação é possível e funcional. Pode observar-se pela figura 5.7 que o programa executa todo o código corretamente, obtendo o fluxo de vídeo da câmara USB ao mesmo tempo que efetua deteções e obtém as leituras do módulo GPS.

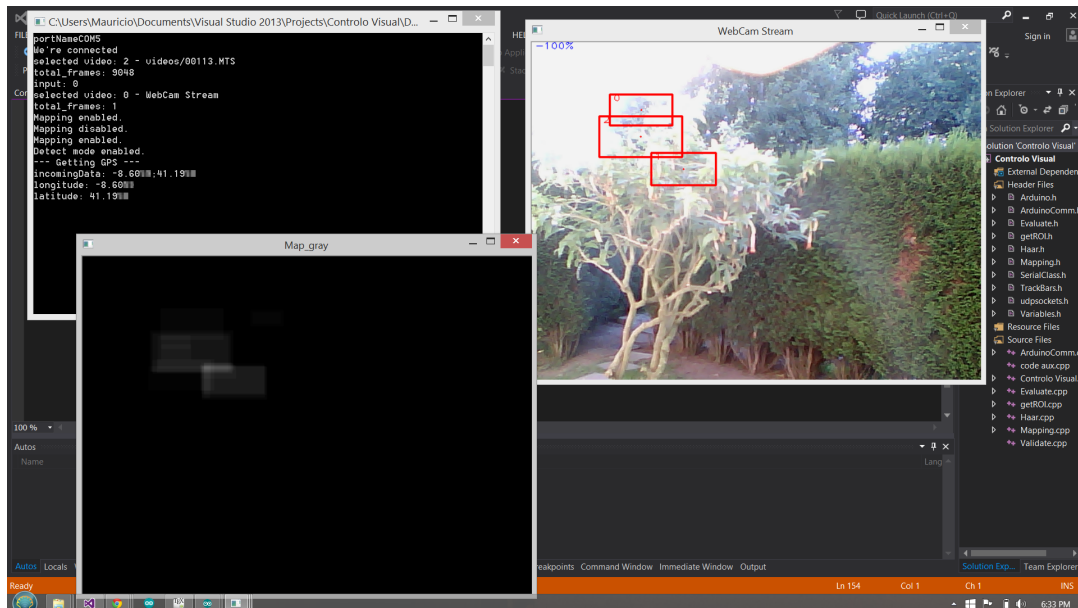


Figura 5.7: Teste de funcionamento da estação de aquisição e aquisição do sinal de GPS.

Neste caso em específico, é possível observar pela janela da linha de comandos, que o detetor e o mapa temporal estão ativos (*enabled*) e que o *Arduino* devolveu a leitura das coordenadas (41,19XX , -8.60XX). Como seria de esperar, temos presentes vários falsos positivos, descartados pelo mapa temporal.

Para concluir, o sistema de aquisição funcionou na perfeição e como seria esperado, faltando apenas a implementação do sensor IMU por protocolo  $I^2C$ .





## Capítulo 6

# Processamento de Imagem

Neste capítulo é apresentado todo o trabalho efetuado no processamento de imagem. Os algoritmos foram desenvolvidos tendo em vista a sua execução em *hardware* dedicado, maximizando a eficiência em tempo-real.

Todo o trabalho foi realizado num computador pessoal através do sistema operativo *Windows 8.1* e da ferramenta *Microsoft Visual Studio 2013* e recorrendo à biblioteca *OpenCV*. Este computador tinha, como especificações técnicas, um processador *Intel® Core™ i7 @1.80 GHz*, 8 *gigabyte* de memória *RAM* e 2 *gigabyte* de memória gráfica dedicada.

### 6.1 Detecção da Linha do Horizonte e Região de Interesse

O primeiro passo no processamento da imagem obtida é a detecção da linha do horizonte, com vista a reduzir a carga computacional exigida posteriormente. O algoritmo de detecção está integrado na função *getROI()*, desenvolvida especificamente para este projeto e responsável por retornar a região de interesse da imagem, sendo esta a região abaixo da linha do horizonte e onde será expectável detetar embarcações.

Uma vez chamada a função *getROI()*, esta começará por converter a imagem passada, de *RGB* para níveis de cinza (*graylevel*). De seguida é aplicada uma limiarização adaptativa através do método de Otsu; foi escolhida a utilização deste método por minimizar a variância intra-classe, maximizando a diferenciação entre pixels de primeiro-plano (*foreground*) e pixels de fundo (*background*) e será explicada a sua adaptatividade posteriormente.

À imagem binária obtida após aplicação do limiar é aplicado o detetor de pixels de fronteira *Canny* cujos valores mínimo e máximo serão, respetivamente, 0 e 255; estes valores não são de crítica escolha dado que o detetor *Canny* é aplicado a uma imagem binária.

De seguida é aplicada a transformada de *Hough* e identificadas todas as linhas, como se observa na figura 6.3.

No entanto, enquanto que este método é extremamente fidedigno em imagens com condições favoráveis, isto é, imagens de dia com luminosidade alta, sol bem acima do plano da imagem e uma clara linha de separação entre mar e céu, observou-se que em condições de contraluz ou em

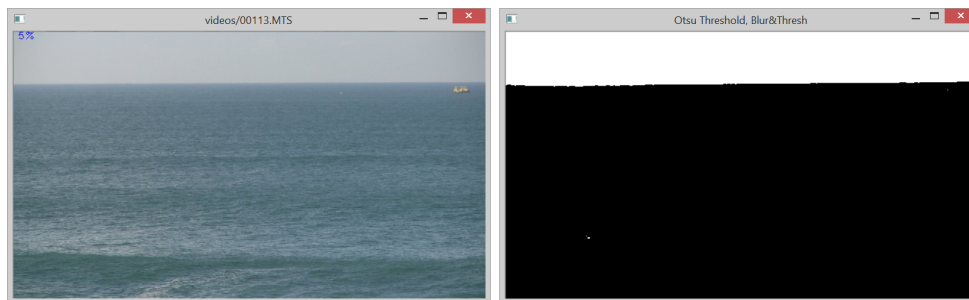


Figura 6.1: Aplicação do limiar por método de Otsu.

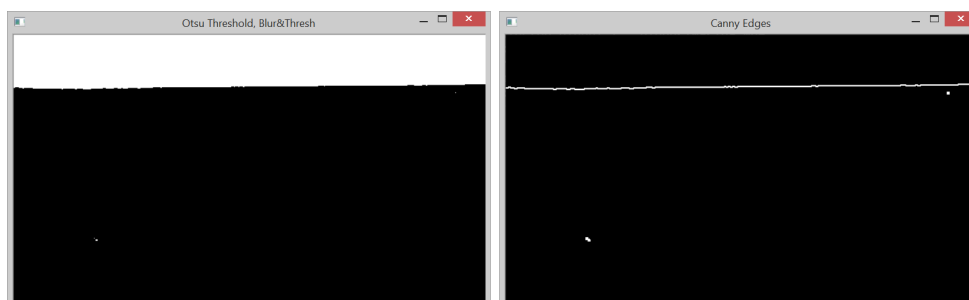


Figura 6.2: Aplicação do detetor de pixels de fronteira Canny

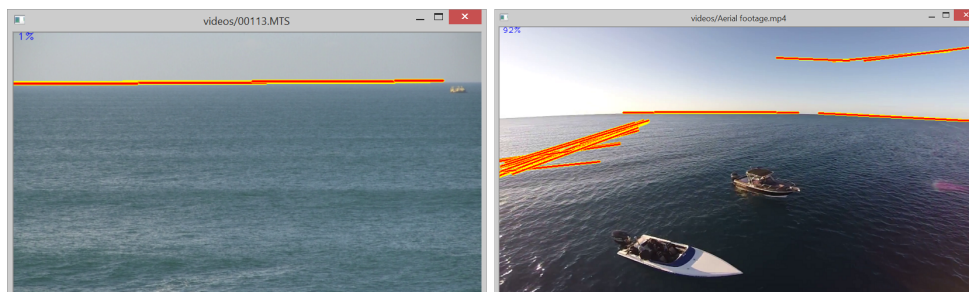


Figura 6.3: Fotogramas analisados e linhas detetadas após transformada de Hough.

imagens onde apenas se observava o mar, este método exigia demasiada carga computacional; isto deve-se ao facto da imagem em questão não ter distintas as duas classes de pixeis - no caso de imagens com apenas mar, as duas classes identificadas são os pixeis do brilho da luz solar refletida na ondulação enquanto a segunda classe são os pixeis do mar. Como tal, com valores de limiar fixos e obtidos pelo método de *Otsu* ter-se-ia um aumento considerável da utilização do *CPU* aquando da tentativa de deteção da linha do horizonte, numa imagem onde tal não era necessário por não existir linha do horizonte. Isto pode ser observado na figura 6.4.

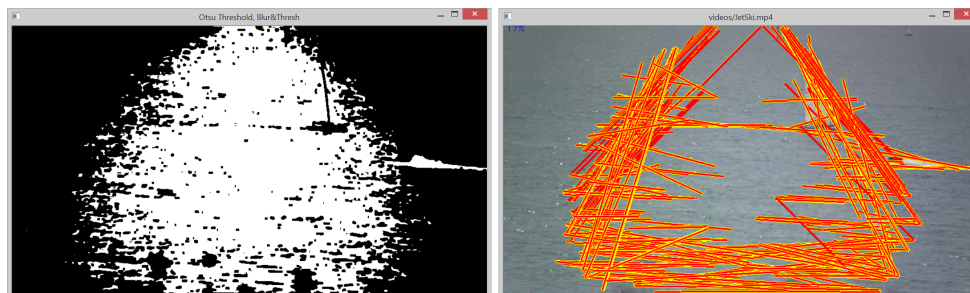


Figura 6.4: Limiarização por método de *Otsu* à esquerda e transformada de *Hough* à direita.

Assim, foi desenvolvido um algoritmo de limiarização adaptativa que consiste num multiplicador ao valor de limiar calculado pelo método de *Otsu*, que incrementa ou decrementa em incrementos 0.1. Desta forma, em imagens onde exista uma clara linha de separação mar-terra, o valor do limiar é perto do calculado enquanto que numa imagem onde são detetadas mais que 5 linhas, deduzindo-se que se trata de uma imagem sem linha do horizonte ou demasiado brilho, o valor do limiar é aumentado através do multiplicador. Desta forma, no mesmo contexto, obtêm-se melhores resultados, observável na figura 6.5.

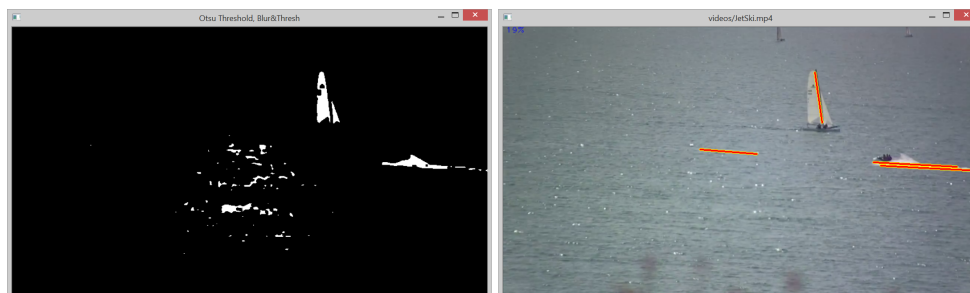


Figura 6.5: Limiar por método de *Otsu* à esquerda e transformada de *Hough* à direita após limiar adaptativo.

Após a identificação das diversas linhas detetadas na imagem analisada é necessário filtrar quaisquer linhas não referentes à verdadeira linha do horizonte. Como tal, são aplicados diversos filtros, desenvolvidos com o intuito de identificar qual das linhas detetadas melhor se aproxima da verdadeira linha do horizonte; isto permite remover falsos positivos, tais como linhas de separação mar-terra, rastros de embarcações, linhas de separação mar-embarcação, onde embarcações de

longo porte geram falsos positivos. O algoritmo responsável pela validação das linhas detetadas começa por remover todas as linhas com um ângulo relativo ao eixo horizontal maior que 5 graus e de seguida cada linha é validada através de um conjunto de pontos pertencentes a duas regiões retangulares, uma imediatamente acima e outra abaixo da linha em análise; se pelo menos 90% dos pontos correspondentes, acima e abaixo da linha, tiverem intensidades diferentes então a linha é classificada como válida e a linha com maior percentagem de disparidade será o candidato com maior probabilidade de ser a verdadeira linha do horizonte (figura 6.6).

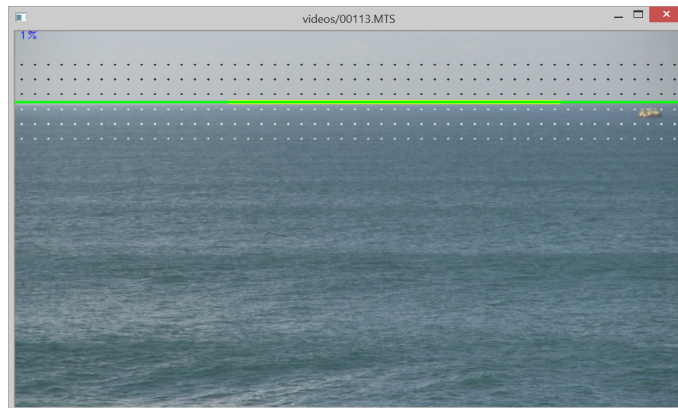


Figura 6.6: *Validação da linha em análise, a amarelo, e linha do horizonte, após validação, a verde.*

Finalmente, a imagem analisada é recortada pela linha anteriormente considerada como a mais válida e retida apenas a parte da imagem imediatamente abaixo da linha do horizonte (figura 6.7).

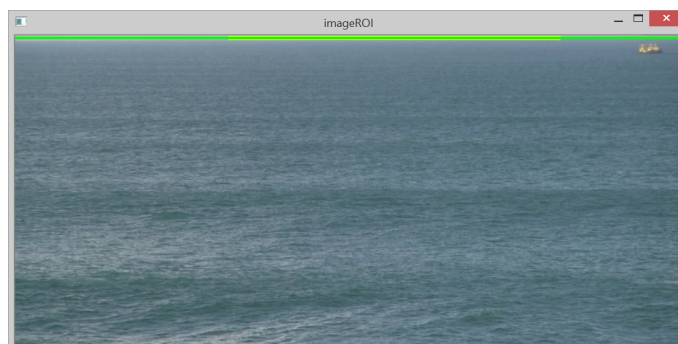


Figura 6.7: *Região de interesse retirada da imagem original.*

## 6.2 Classificador em cascata com características *Haar*

Com vista a detetar as embarcações presentes nas imagens, foi desenvolvido um algoritmo de deteção de embarcações apoiado em métodos de deteção de objetos desenvolvidos originalmente no campo da visão computacional.

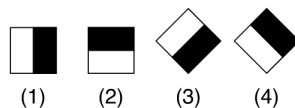
Embora existam vários algoritmos diferentes para detecção de objetos, cada um tem as suas vantagens e desvantagens. Uma imagem é uma coleção de cor e/ou intensidade luminosa e analisar cada pixel para detecção de embarcações é demorado e difícil de obter devido à vasta variedade de cores e formas das embarcações. Neste caso, e tratando-se da detecção de objetos com características variadas e não seguindo um padrão, a maioria peca em obter resultados favoráveis, seja por operações morfológicas, extração de cor ou algoritmos de subtração de fundo.

Foi, então, desenvolvido um detetor, baseado no *framework* de detecção de objetos originalmente desenvolvido para detecção de faces, chamado "Haar Classifiers", para detecção rápida de qualquer objeto utilizando para isso uma cascata de classificadores gerados pelo algoritmo *Ada-Boost* [34]. O núcleo deste detetor é um classificador binário que distingue imagens de embarcações de não-embarcações, fazendo para isso um varrimento da imagem com uma sub-janela de dimensão variável.

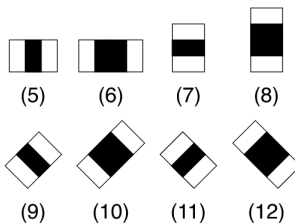
### 6.2.1 Características Haar

O classificador utiliza as características *Haar* e as características Haar rodadas [49]. As características *Haar* são características simples baseadas na diferença da soma do valor de intensidade dos pixels em duas zonas da imagem. Na figura 6.8 observam-se as características *Haar* utilizadas.

Características de Fronteira



Características de Linha



Características de Centro Rodeado

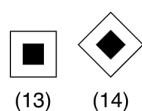


Figura 6.8: Características Haar utilizadas.

Estas características, ao invés de utilizarem os valores de intensidade de cada pixel, utilizam a variação dos valores de contraste entre grupos retangulares de pixels adjacentes. A variância do contraste entre os grupos de pixels permite determinar as áreas brancas e pretas; dois ou três grupos adjacentes com uma variância de contraste relativa formam uma característica *Haar*. Estas

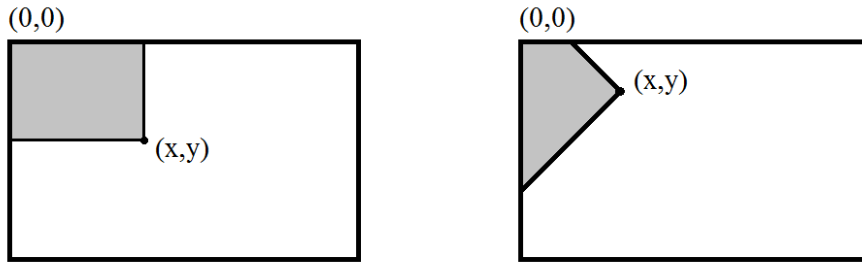
características podem ser, facilmente, redimensionadas aumentando ou diminuindo o tamanho do grupo de pixels a analisar permitindo que estas características sejam utilizadas para detetar objetos de qualquer dimensão.

Para o cálculo das características de uma imagem é utilizada uma representação intermédia da imagem em análise, denominada imagem integral [34]. A imagem integral é uma matriz cujos valores são a soma dos valores de intensidade dos pixels diretamente à esquerda e acima de um pixel na posição  $(x, y)$ , inclusive. Assim, se  $A[x,y]$  representar a imagem original e  $AI[x,y]$  representar a imagem integral, então esta é calculada como demonstrado na equação 6.1 e ilustrado na figura 6.9a.

$$AI[x,y] = \sum_{x' \leq x, y' \leq y} A(x', y') \quad (6.1)$$

As características inclinadas 45 graus, como a característica de linha na figura 6.8 (3), tal como apresentado por Lienhart e Maydt, requerem uma outra representação intermédia denominada imagem integral rodada [49]. A imagem integral rodada é calculada através da soma dos valores de intensidade dos pixels que estão a 45 graus à esquerda e acima do valor de  $x$  e abaixo do valor de  $y$ . Se  $A[x,y]$  representar a imagem original e  $AR[x,y]$  representar a imagem integral rodada, então, a imagem integral é calculada como apresentado na equação 6.2 e ilustrado na figura 6.9b.

$$AR[x,y] = \sum_{x' \leq x, y' \leq x - |y - y'|} A(x', y') \quad (6.2)$$



(a) Área somada de uma imagem integral. (b) Área somada de uma imagem integral rodada.

Figura 6.9: Áreas da soma de imagens integrais.

São necessários apenas 2 passos para calcular ambas as imagens integrais, um para cada matriz. Recorrendo à matriz apropriada e através da diferença entre 6 a 8 elementos, que formam 2 ou 3 áreas retangulares conectadas, uma característica de qualquer dimensão pode ser calculada; assim, calcular uma característica é extremamente rápido e eficiente e o cálculo de características de dimensões variadas exige o mesmo tempo de processamento que uma característica de dois ou três pixels. Isto significa que a deteção de embarcações de várias dimensões impõe a mesma carga computacional e exige o mesmo tempo que a deteção de embarcações da mesma dimensão dado

que o dimensionamento não requer esforço adicional [34]. Apesar da capacidade de processamento da estação de processamento ser elevada, esta abordagem torna possível adaptar o sistema de aquisição e processamento num só, sendo possível efetuar um processamento local e tornar este sistema portátil.

O cálculo de uma determinada área de uma característica pode ser facilmente obtida através da equação 6.3, onde os pontos  $A, B, C, D$  pertencem à imagem integral  $I$ , como ilustrado na figura 6.10.

$$Soma = I(C) + I(A) - I(B) - I(D) \quad (6.3)$$

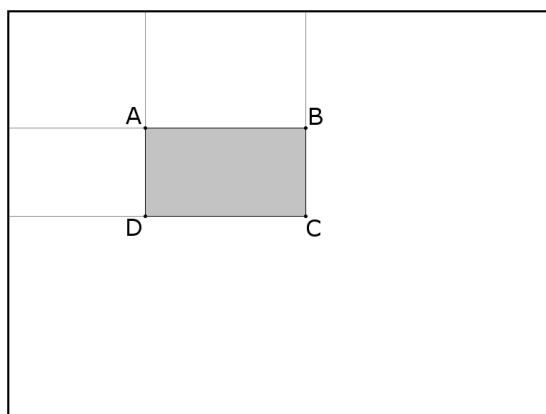


Figura 6.10: Cálculo da área sombreada de uma imagem integral.

### 6.2.2 AdaBoost

As características *Haar* são um classificador fraco, tomam uma decisão um pouco melhor que pelo acaso (0.5 de probabilidade). O algoritmo *Gentle AdaBoost* descrito em [50] combina múltiplas características *Haar* criando um classificador forte. Este algoritmo é uma variante da robusta técnica amplificada de aprendizagem [51] e é utilizado para escolher um conjunto de árvores de classificação e regressão (*CART's*) para alcançar uma taxa de deteção e taxa de erro estipuladas.

O algoritmo *Gentle AdaBoost* combina *CART's* onde as características são a entrada  $(x)$ ,  $y \in [-1; 1]$  é a classificação da característica (cuja polaridade indica se é positiva (isto é, se contém o objeto alvo) ou negativa (não contém o objeto alvo) e a saída é a probabilidade:

$$f_m(x) = P_w(y = 1|x) - P_w(y = -1|x) \quad (6.4)$$

A aprendizagem é baseada em  $N$  exemplos de treino ponderados  $(x_1, y_1), \dots, (x_N, y_N)$ ; segue-se o algoritmo de treino, onde  $N$  é o número de amostras,  $M$  é o número de classificadores fracos:

1. Inicializar os pesos  $w_i$  com  $1/N, i = 1, 2, \dots, N$  e o classificador  $F(x) = 0$ ;



2. Para  $m=1,2,\dots,M$ :

- (a) Ajustar a função de regressão da árvore de decisão  $f_m(x)$  através dos mínimos quadrados ponderados de  $y_i$  a  $x_i$  com pesos  $w_i$ ;
- (b) Atualizar  $F(x) = F(x) + f_m(x)$ ;
- (c) Atualizar  $w_i = w_i e^{-y_i f_m(x)}$  e normalizar tal que:

$$\sum_{m=1}^N w_i = 1. \quad (6.5)$$

São adicionados classificadores fracos até que o classificador atinja as taxas de detecção e de falsos positivos estipuladas.

3. O classificador resultante é  $\text{sign}[F(x)]$  onde:

$$F(x) = \sum_{m=1}^M f_m(x). \quad (6.6)$$

### 6.2.3 Classificadores em Cascata

O desempenho de um único classificador não é adequado para uma correta classificação de objetos, dado que produz uma taxa de sucesso alta, por exemplo 0,995, mas também uma alta taxa de erro, por exemplo 0,5. No entanto, a taxa de sucesso é consideravelmente superior à taxa de erro. Para uma construção de um classificador robusto, vários classificadores fracos são dispostos em cascata, isto é, como uma árvore de decisão degenerada; em cada etapa da cascata, é tomada a decisão se a sub-imagem em análise contém o objeto alvo ou não; a sub-janela que é validada pela última etapa é classificada como um objeto no final. Esta computação reduz ambas as taxas, dado que a taxa de sucesso é perto de 1 a sua multiplicação resulta num resultado perto de 1 enquanto que a multiplicação das taxas de erro reduzidas se aproxima de 0. Além disso, isto acelera todo o processo de classificação dado que grandes partes da imagem que não contêm dados relevantes são descartadas rapidamente nas primeiras etapas, como por exemplo mar homogêneo, enquanto que as sub-janelas mais difíceis, por exemplo rastros de embarcações ou rebentação das ondas, são classificadas como não sendo embarcações nas etapas seguintes.

Uma cascata eficaz é facilmente aprendida por um método iterativo simples; para cada etapa, a função classificadora  $h(x)$  é aprendida até ser alcançada a desejada taxa de sucesso. O processo continua e a etapa seguinte usa os exemplos positivos corretamente classificados (sub-imagens que contêm embarcações) e os exemplos negativos (sub-imagens sem embarcações contidas) erroneamente classificados até à data. Estes exemplos negativos são partes de imagens aleatórias geradas a partir do conjunto de exemplos negativos fornecidos que são aprovados pelas etapas anteriores e são, desta forma, erroneamente classificados. Na figura 6.11 observa-se um exemplo de um classificador em cascata para detecção de um objeto em imagens 2D.



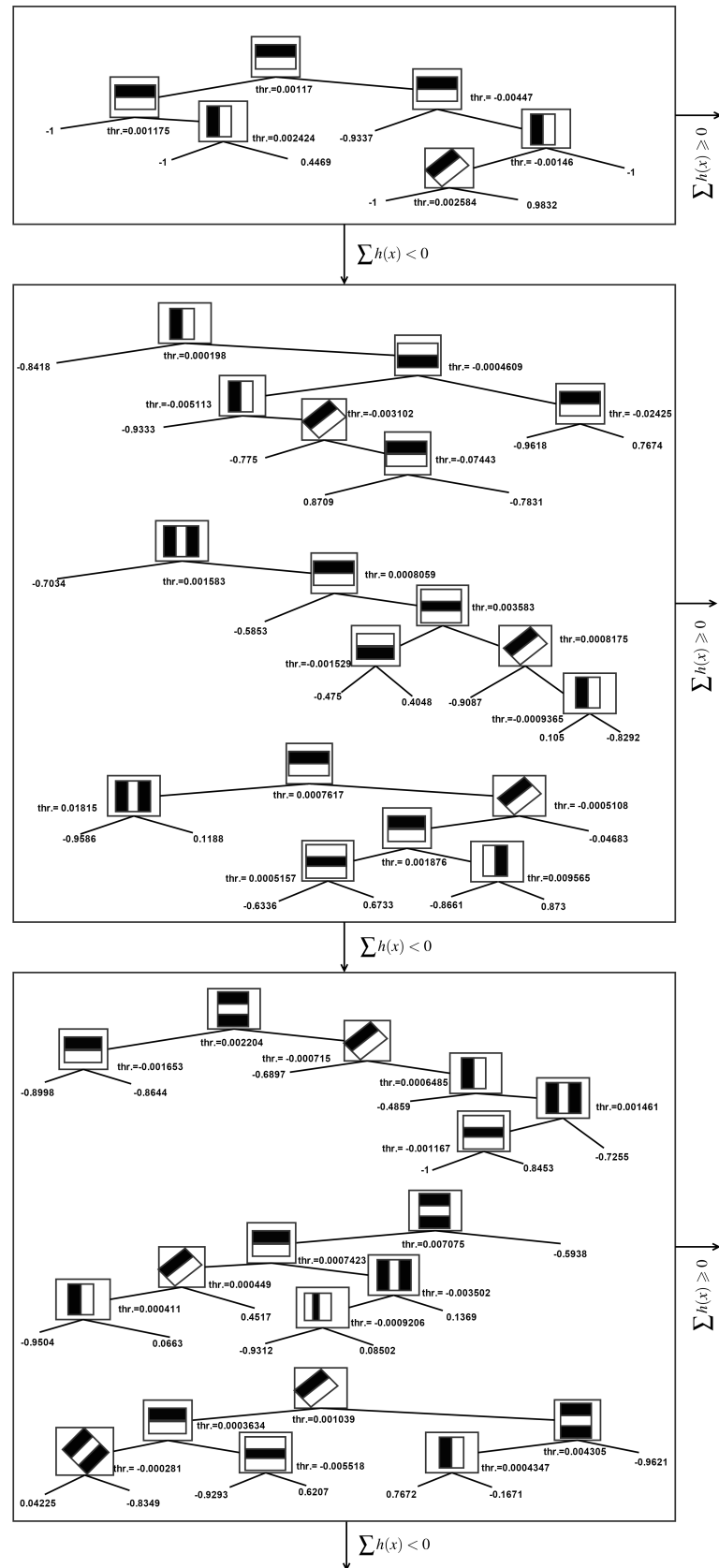


Figura 6.11: As primeiras 3 etapas de um classificador em cascata. Cada etapa contém várias árvores de classificação simples que utilizam as características Haar e um limiar "thr" e retornam  $\sum h(x)$ , determinado através do percurso pelas árvores.

A taxa de falsos positivos da cascata final é:

$$F = \prod_{i=1}^K f_i, \quad (6.7)$$

onde  $K$  é o número de etapas,  $f_i$  é a taxa de falsos positivos do classificador da etapa  $i$ . A taxa de detecção é dada por:

$$D = \prod_{i=1}^K d_i, \quad (6.8)$$

onde  $d_i$  é a taxa de detecção do classificador da etapa  $i$ . Assim, um classificador com 20 etapas,  $d_i = d = 0.995$  e  $f_i = f = 0.5$  resulta numa taxa de detecção teórica  $D \approx 0.9$  e numa taxa de erro  $F \approx 9.5 \times 10^{-7}$ .

## 6.3 Treino do Classificador

O treino do classificador foi realizado recorrendo a funções disponibilizadas pela biblioteca *OpenCV*. Assim, apenas foi necessário fazer a preparação dos dados de entrada para o algoritmo de treino. Os dados necessários foram as imagens positivas, isto é, imagens nas quais estão presentes embarcações que se queriam detetar e imagens negativas, nas quais não está presente qualquer tipo de embarcação e são encaradas como imagens de fundo. Dada a abordagem desta dissertação, optou-se por treinar o classificador com o máximo de positivos possível, nos quais as embarcações são de diversos tipos, em diferentes poses e em condições variadas. Foram utilizadas 431 imagens positivas e 1990 imagens negativas.

### 6.3.1 Positivos

Os exemplos positivos obtidos para o treino do classificador foram manualmente obtidos através de pesquisas de imagens na *web*, assim como extraídos manualmente de vídeos obtidos também através de pesquisas *online*. No entanto, estes exemplos são de dimensões e resoluções diversas; como tal, foi necessário cortar os exemplos positivos de forma a que estes contivessem a maior área possível da embarcação e a menor área possível de fundo. Efetuar esta operação manualmente, em programas de edição de imagem, demoraria demasiado tempo, pelo que se recorreu a uma marcação das embarcações nos exemplos positivos.

Esta marcação foi realizada através de um programa de marcação de imagens, facilmente obtido na *internet*. Após a marcação manual de todas as embarcações em cada exemplo positivo, é gerado um vetor, com a localização das embarcações marcadas em cada imagem, que será posteriormente utilizado no treino do classificador. Na figura 6.12 estão presentes alguns exemplos positivos. A figura 6.13 mostra um exemplo da marcação de uma embarcação num dos exemplos positivos.



Figura 6.12: *Exemplos positivos de embarcações.*



Figura 6.13: *Marcação de uma embarcação num exemplo positivo.*

Finalmente, é gerado o vetor de exemplos positivos utilizado no treino do classificador através da função *opencv\_createsamples* disponibilizada, também, pela biblioteca OpenCV. A função é responsável por converter os exemplos positivos (pós-marcação) para níveis de cinza e redimensioná-los para  $40 \times 20$  pixels. No *framework* proposto por [34], a dimensão aconselhada será  $20 \times 20$  pixels, a qual foi adaptada para o valor utilizado de  $40 \times 20$  pixels para acomodar as dimensões mais frequentes das embarcações nos exemplos positivos que, regra geral, serão sempre mais compridas que altas. Este vetor será o responsável por indicar ao algoritmo de treino as coordenadas dos exemplos positivos em cada imagem marcada para posterior processamento.

### 6.3.2 Negativos

Os exemplos negativos foram maioritariamente extraídos de vídeos obtidos por pesquisa *on-line*, nomeadamente de vídeos de paisagens e mar aberto sem embarcações. A opção foi retirar o máximo de imagens de fundo o mais semelhantes possível às condições onde seria expectável observar embarcações, isto é, mar aberto em condições meteorológicas diversas, zonas de rebentação do mar, zonas costeiras com ou sem edifícios e rastos de embarcações. Todos os exemplos negativos, provenientes de vídeos, foram redimensionados para  $1280 \times 720$  pixels e convertidos para escala de cinza enquanto que outras imagens foram redimensionadas para uma altura de 720 pixels. De seguida foi criado um ficheiro de texto contendo o nome de todos os exemplos negativos que foi passado como argumento ao algoritmo de treino servindo de apontador para os exemplos negativos. Na figura 6.14 observam-se alguns exemplos negativos.



Figura 6.14: Alguns exemplos negativos utilizados.

### 6.3.3 Aprendizagem

O processo de aprendizagem foi realizado recorrendo à função *opencv\_traincascade* cujos parâmetros de entrada foram o vetor de exemplos positivos, o ficheiro de texto apontador para os exemplos negativos, a taxa de sucesso, a taxa de falsos positivos, a dimensão dos exemplos de treino ( $40 \times 20$ ) pixels, entre outros. Foram realizados diversos treinos, resultando em diversos classificadores, variando, por exemplo, o número de exemplos positivos ou utilizando imagens após equalização mas principalmente aumentando o número de etapas entre 16 e 24. Os resultados e comparações entre cada classificador podem ser observados no sub-capítulo 6.4.

```

C:\Windows\system32\cmd.exe - opencv_traincascade -data cascades ...
PARAMETERS:
cascadeDirName: cascades
vecFileName: vector/vector431p.vec
bgFileName: negative/bg.txt
numPos: 387
numNeg: 1990
numStages: 16
precalcValBufSize[Mb]: 1024
precalcIdxBufSize[Mb]: 2048
stageType: BOOST
featureType: HOG
sampleWidth: 40
sampleHeight: 20
boostType: GBDT
minHitRate: 0.9995
maxFalseAlarmRate: 0.5
weightTrimRate: 0.95
maxDepth: 1
maxWeakCount: 100
mode: ALL

===== TRAINING 0-stage =====
BEGIN
POS count : consumed 387 : 387
NEG count : acceptanceRatio 1990 : 1
Precalculation time: 46.773
+-----+
| N | HR | FA |
+-----+
| 1 | 1 | 1 |
+-----+
| 2 | 1 | 1 |
+-----+
| 3 | 1 | 1 |
+-----+

(a) Início de treino com 16 etapas.

C:\Windows\system32\cmd.exe
| 21 | 1 | 0.463921
+-----+
END
Training until now has taken 0 days 6 hours 16 minutes 50 seconds.

===== TRAINING 19-stage =====
BEGIN
POS count : consumed 387 : 387
NEG count : acceptanceRatio 1990 : 0.80245219
Precalculation time: 47.813
+-----+
| N | HR | FA |
+-----+
| 1 | 1 | 1 |
+-----+
| 2 | 1 | 1 |
+-----+
| 3 | 1 | 1 |
+-----+
| 4 | 1 | 0.9873371 |
+-----+
| 5 | 1 | 0.9090451 |
+-----+
| 6 | 1 | 0.8909551 |
+-----+
| 7 | 1 | 0.808041 |
+-----+
| 8 | 1 | 0.8095481 |
+-----+
| 9 | 1 | 0.7663321 |
+-----+
| 10 | 1 | 0.7211061 |
+-----+
| 11 | 1 | 0.6653271 |
+-----+
| 12 | 1 | 0.6728641 |
+-----+
| 13 | 1 | 0.5683421 |
+-----+
| 14 | 1 | 0.5532661 |
+-----+
| 15 | 1 | 0.6457291 |
+-----+
| 16 | 1 | 0.5035181 |
+-----+
| 17 | 1 | 0.4532661 |
+-----+
END
Training until now has taken 0 days 6 hours 35 minutes 33 seconds.
  
```

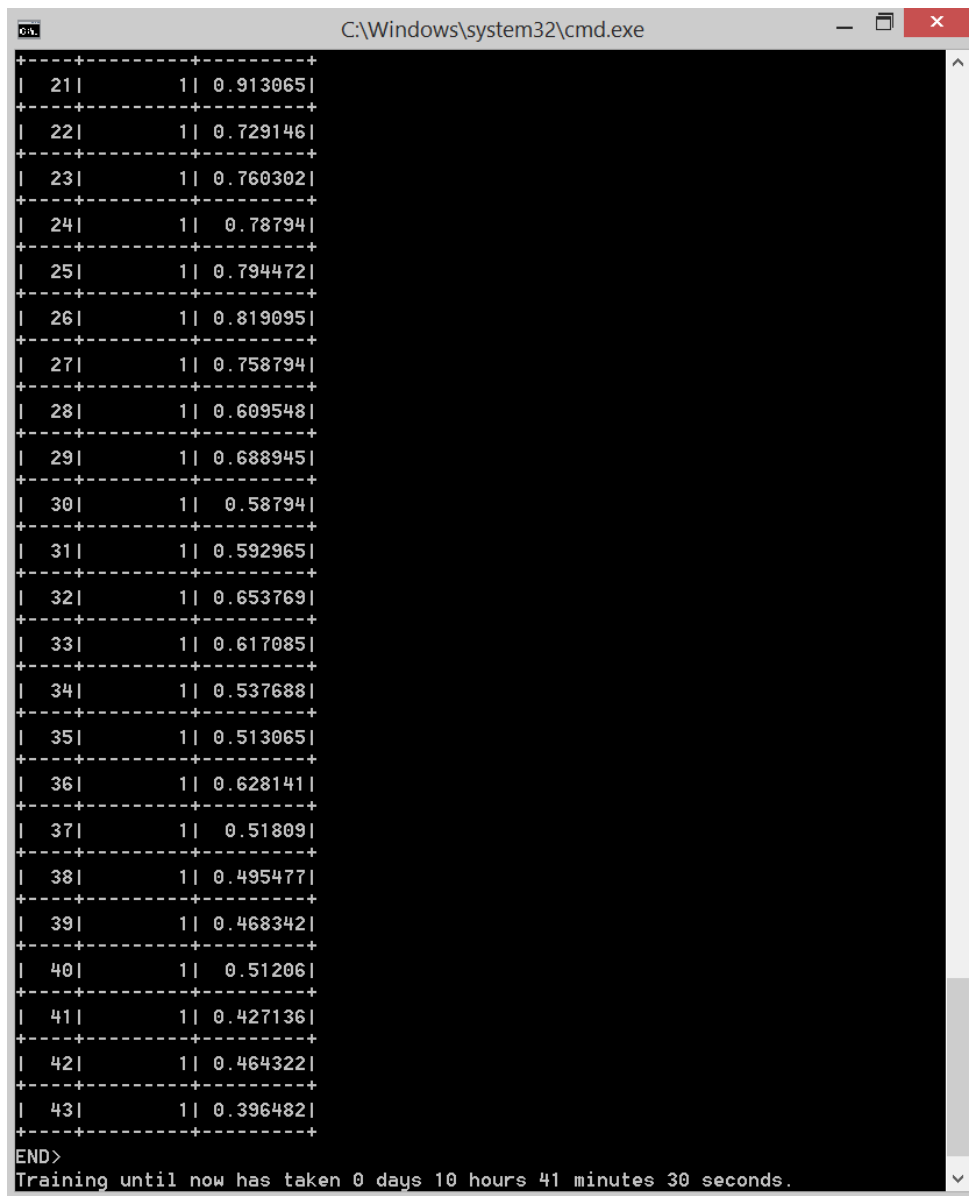
(a) Início de treino com 16 etapas.

(b) Última de 20 etapas de treino.

Figura 6.15: Treino de um classificador com 16 etapas (6.15a) e de um classificador com 20 etapas (6.15b).

Em ambos os cenários, foi utilizado o vetor com 431 exemplos positivos, dos quais 90%, ou seja 387, foram admitidos como o número de exemplos positivos; isto deve-se ao facto de, no caso de serem mal identificados alguns exemplos positivos como negativos, ser possível utilizar na mesma 387 exemplos positivos indo buscar um exemplo positivo ainda não utilizado no treino. Foram utilizados 1990 exemplos negativos e em ambos os casos admitidas taxas de sucesso de 0,995 e taxas de falsos positivos de 0,5. Observa-se, também, que na última de 20 etapas foram utilizadas 17 características para se atingir a taxa de falsos positivos estipulada ( $0.453266 < 0.5$ ).

Como será demonstrado no sub-capítulo 6.4, os melhores resultados foram obtidos utilizando um classificador com 20 etapas e taxa de falsos positivos de 0,4; de forma a obter esta taxa de falsos positivos, o treino deste classificador necessitou de calcular mais características (figura 6.16)



```
C:\Windows\system32\cmd.exe

+-----+
| 21|      1| 0.913065|
+-----+
| 22|      1| 0.729146|
+-----+
| 23|      1| 0.760302|
+-----+
| 24|      1| 0.78794|
+-----+
| 25|      1| 0.794472|
+-----+
| 26|      1| 0.819095|
+-----+
| 27|      1| 0.758794|
+-----+
| 28|      1| 0.609548|
+-----+
| 29|      1| 0.688945|
+-----+
| 30|      1| 0.58794|
+-----+
| 31|      1| 0.592965|
+-----+
| 32|      1| 0.653769|
+-----+
| 33|      1| 0.617085|
+-----+
| 34|      1| 0.537688|
+-----+
| 35|      1| 0.513065|
+-----+
| 36|      1| 0.628141|
+-----+
| 37|      1| 0.51809|
+-----+
| 38|      1| 0.495477|
+-----+
| 39|      1| 0.468342|
+-----+
| 40|      1| 0.51206|
+-----+
| 41|      1| 0.427136|
+-----+
| 42|      1| 0.464322|
+-----+
| 43|      1| 0.396482|
+-----+
END>
Training until now has taken 0 days 10 hours 41 minutes 30 seconds.
```

Figura 6.16: Última de 20 etapas de um classificador com taxa de falsos positivos = 0,4, utilizando para isso 43 classificadores fracos.



## 6.4 Resultados da Detecção

Após o treino dos vários classificadores, foi então implementado o algoritmo de detecção de embarcações, com recurso a mais uma função da biblioteca *OpenCV*, *detectMultiScale()*; esta faz um registo de todas as deteções num vetor de retângulos, no qual cada retângulo, guardado no vetor com as suas coordenadas na imagem analisada, é uma deteção. Daí se obtêm os resultados ilustrados nas figuras 6.17, 6.18 e 6.19, respetivamente para classificadores com 16, 18 e 20 etapas.

É possível observar-se que com o aumento do número de etapas, o número de falsos positivos é diminuído, mantendo-se o número de positivos corretamente detetados. A não deteção das embarcações mais longínquas nas figuras 6.17a, 6.18a e 6.19a é devido ao facto da linha de horizonte ser detetada e a imagem ser cortada; isto seria facilmente resolvido elevando o plano da câmara, o que não foi possível na altura de aquisição dos vídeos em análise. Em relação às figuras 6.17b, 6.18b e 6.19b, as embarcações mais longínquas não são detetadas devido ao seu pequeno porte e elevada distância à plataforma de aquisição, assim como devido à compressão das imagens a analisar, reduzindo drasticamente a informação nelas contida. Finalmente, nas imagens 6.17c, 6.18c e 6.19c são detetadas todas as embarcações, embora a embarcação mais à direita não seja totalmente detetada, devido sobretudo à falta de exemplos positivos de navios. Foi também treinado um classificador com 24 etapas que ficou sobre-ajustado, não sendo capaz de efetuar nenhuma deteção. Desta forma, como o classificador em cascata com 20 etapas apresentou resultados ótimos, foi tomada a decisão de o considerar como o mais fidedigno.

Relativamente aos tempos de processamento, com a execução de todos os algoritmos implementados, o computador utilizado nesta dissertação conseguiu fazer uma leitura média de 4 *frames/s*, sendo que, em imagens onde a linha do horizonte era detetada e a imagem posteriormente cortada, se chegou a obter 10 *frames/s*; no entanto, em imagens onde, na totalidade da mesma, era aplicado o detetor obteve-se, por vezes, 2 *frames/s*, limitando o desempenho em tempo-real deste sistema.

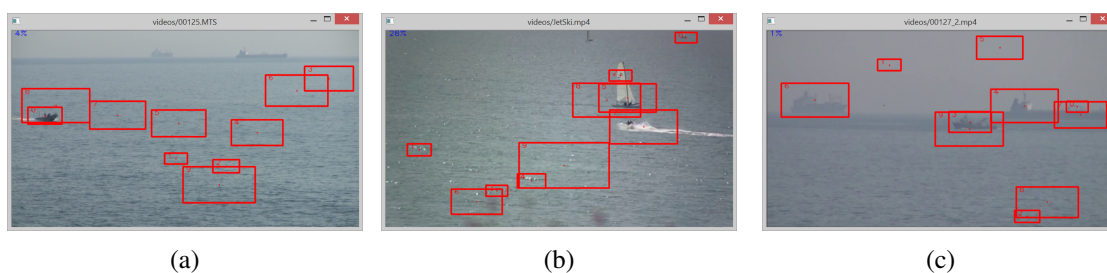


Figura 6.17: *Classificador em cascata com 16 etapas.*

A utilização de um classificador com 20 etapas traduziu-se em resultados mais favoráveis e mais fidedignos. Como termo de comparação, foram aplicados ao mesmo vídeo, os 4 classificadores acima documentados e efetuada uma comparação, como se pode observar nos gráficos das figuras 6.20, 6.21, 6.22 e 6.23. O vídeo utilizado para a comparação é o das figuras 6.17c, 6.18c e 6.19c e tem presentes, na totalidade dos fotogramas analisados, as mesmas 3 embarcações.

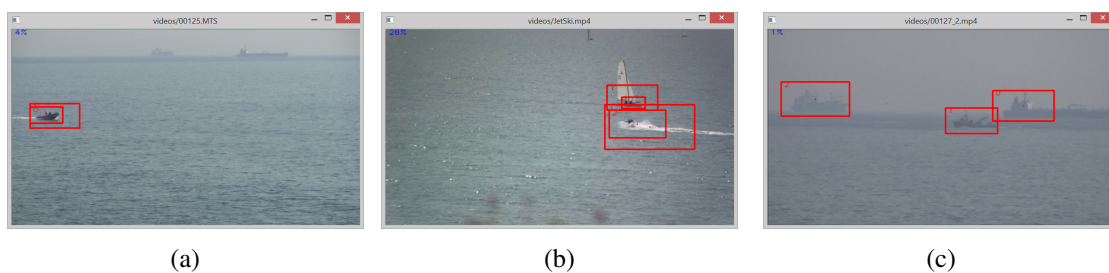


Figura 6.18: *Classificador em cascata com 18 etapas.*

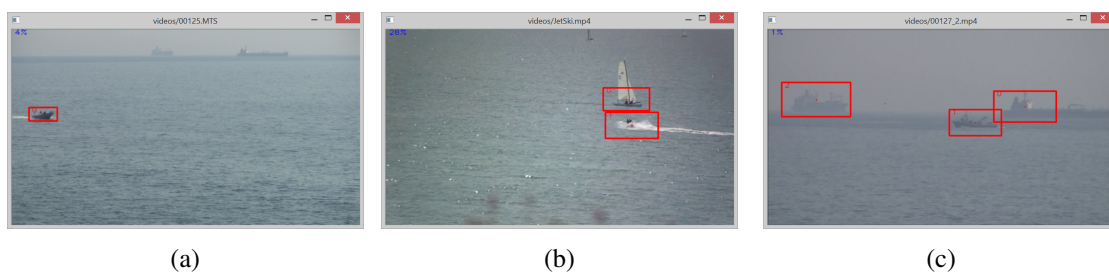


Figura 6.19: *Classificador em cascata com 20 etapas.*



Figura 6.20: *Gráfico de detecções com um classificador em cascata com 16 etapas para o vídeo 00127.MTS com 3 embarcações.*

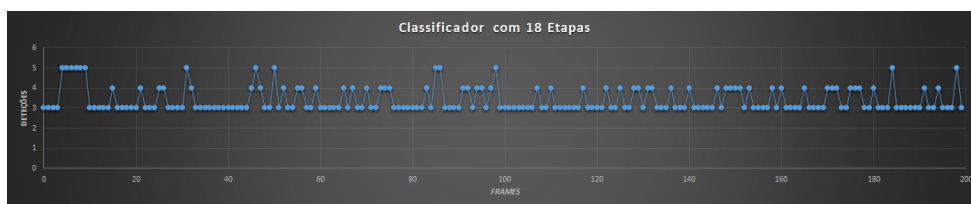


Figura 6.21: *Gráfico de detecções com um classificador em cascata com 18 etapas para o vídeo 00127.MTS com 3 embarcações.*

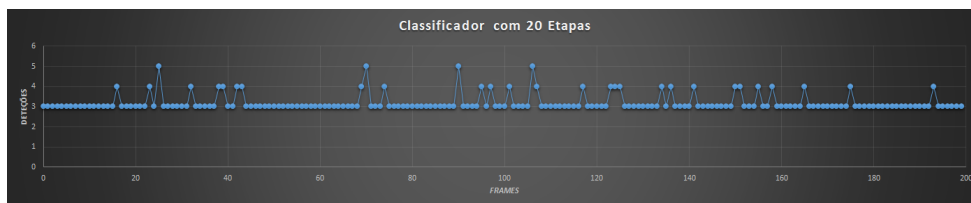


Figura 6.22: *Gráfico de detecções com um classificador em cascata com 20 etapas para o vídeo 00127.MTS com 3 embarcações.*



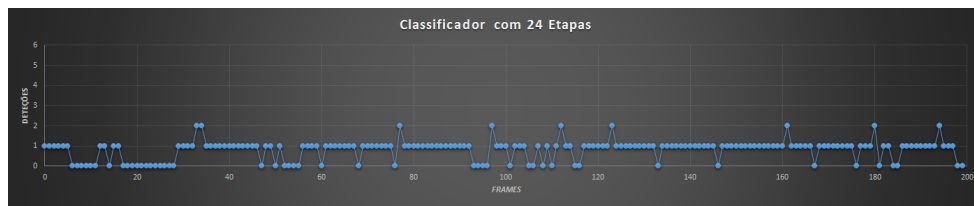


Figura 6.23: Gráfico de detecções com um classificador em cascata com 24 etapas para o vídeo 00127.MTS com 3 embarcações.

É facilmente observável a coerência e consistência do classificador com 20 etapas, sendo o que mais vezes detetou apenas as 3 embarcações; a maior parte das detecções erradas deste classificador foram sobreposições à mesma embarcação enquanto que nos classificadores com menos etapas foram detecções ruidosas. Notória, também, a falha na detecção de embarcações do classificador com 24 etapas, onde nunca foi capaz de detetar as 3 embarcações, mostrando assim o seu sobre-ajuste.

A tabela 6.1, mostra a média mas, mais importante, mostra também a variância de cada classificador, para uma comparação válida.

Tabela 6.1: Tabela de comparação entre os classificadores treinados, para 200 fotografias.

Classificador	Média de Detecções	Variância
16 Etapas	6.020	3.005
18 Etapas	3.417	0.4020
<b>20 Etapas</b>	<b>3.191</b>	<b>0.1759</b>
24 Etapas	0.8090	2.206

Observa-se que os classificadores com 18 e 20 etapas apresentam uma média aproximada, com uma diferença de apenas  $\approx 7,5\%$ ; no entanto, a variância do classificador com 20 etapas é consideravelmente inferior, fazendo do mesmo a melhor escolha para aplicar em detecção de embarcações.

Foram também realizados treinos e testes destes mesmos classificadores com imagens equalizadas, isto é, utilizando exemplos negativos equalizados no treino dos classificadores, e equalizando as imagens no pré-processamento anterior à aplicação do detetor. No entanto, os resultados obtidos foram menos favoráveis que o expectável, como se pode observar pelos gráficos das figuras 6.24 e 6.25.

Tabela 6.2: Tabela de comparação entre os classificadores treinados, com imagens equalizadas, para 200 fotografias.

Classificador com Equalização	Média de Detecções	Variância
18 Etapas	3.111	0.6583
20 Etapas	1.275	1.740

Pela tabela 6.2, observa-se que, apesar do classificador com 18 etapas e imagens equalizadas

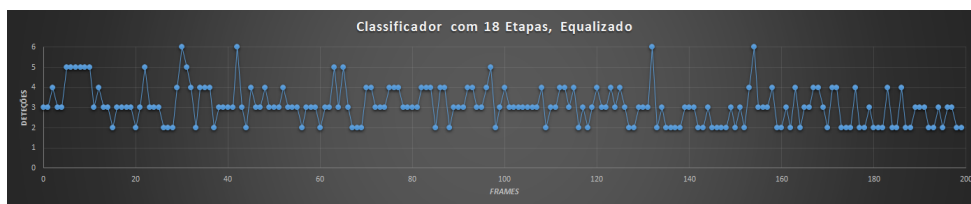


Figura 6.24: Gráfico de detecções com um classificador em cascata com 20 etapas e imagens equalizadas, para o vídeo 00127.MTS com 3 embarcações.

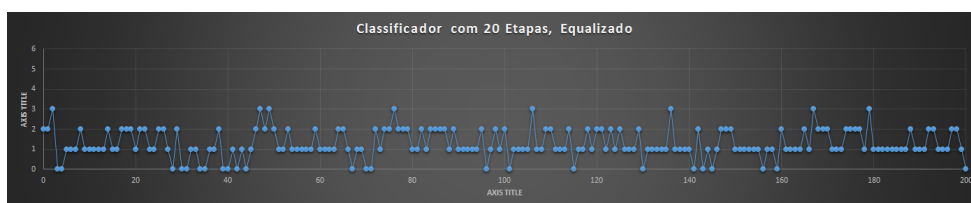


Figura 6.25: Gráfico de detecções com um classificador em cascata com 24 etapas e imagens equalizadas, para o vídeo 00127.MTS com 3 embarcações.

apresentar uma média mais perto da desejada que o classificador de 20 etapas anterior, apresenta, no entanto, uma variância superior, não só à do classificador de 20 etapas mas também, à do classificador de 18 etapas não equalizado. Com isto se pode deduzir que, dados os parâmetros em que esta dissertação foi desenvolvida, a utilização de imagens equalizadas produzirá resultados abaixo do nível ótimo, conseguido sem equalização. A equalização do histograma apresenta melhores resultados quando os planos de fundo e de primeiro plano são ambos escuros ou claros, o que não se verifica nas imagens utilizadas nesta dissertação; também traz repercussões indesejáveis quando aplicado a imagens com baixa profundidade de cor, isto é, se aplicado a imagens de *8-bit* mostradas através uma paleta de níveis de cinza com *8-bit*, reduzirá ainda mais a profundidade de cor da imagem. Obtiveram-se, por isso, piores resultados com imagens equalizadas.

## 6.5 Mapa Temporal

Após a aplicação do detetor de embarcações, foi necessário validar as detecções obtidas, de forma a que embarcações detetadas com sucesso na imagem sejam corretamente registadas e de forma a que os falsos positivos sejam imediatamente descartados. Isto foi conseguido através da utilização de um mapa temporal, capaz de registar a variação, no tempo, das detecções efetuadas; assim tem-se acesso a um registo temporal que torna possível validar as embarcações com base na consistência e coerência de uma deteção ao longo do tempo.

O mapa temporal é, nada mais que, uma matriz da mesma dimensão que a imagem em análise, inicializada a 0 no início do programa. Quando é efetuada uma deteção, o programa executa o algoritmo de validação que começa por atualizar o mapa; esta atualização é efetuada incrementando os valores da matriz apenas nas áreas onde foram detetadas embarcações, ou seja, no interior dos

retângulos resultantes do detetor, e decrementando os valores nas restantes áreas. Isto resulta num mapa com maioritariamente valores nulos (representados a preto) e áreas com valores não nulos (representados em tons de cinza e a branco) representativos da confiança na deteção efetuada nessa área; quanto maiores estes valores não nulos, mais confiança se apresenta nessa área e mais probabilidade existe de se tratar de uma embarcação correta e consistentemente detetada. Os valores de incrementação e decrementação estipulados foram, respectivamente,  $+3$  e  $-6$ , sendo que o valor máximo possível de obter é 255. Na figura 6.26 é possível observar vários exemplos de deteções; a vermelho (6.27a) as deteções detetadas recentemente, a amarelo (6.27b) as deteções perto de serem consideradas válidas e finalmente, a verde (6.27c), as deteções consideradas válidas.

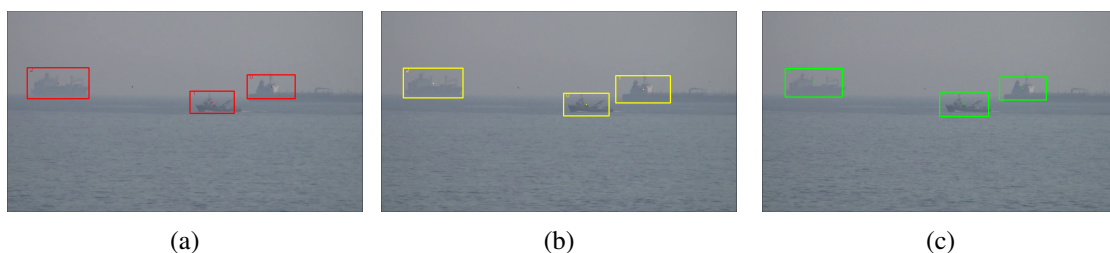


Figura 6.26: *Deteção e Validação em execução.*

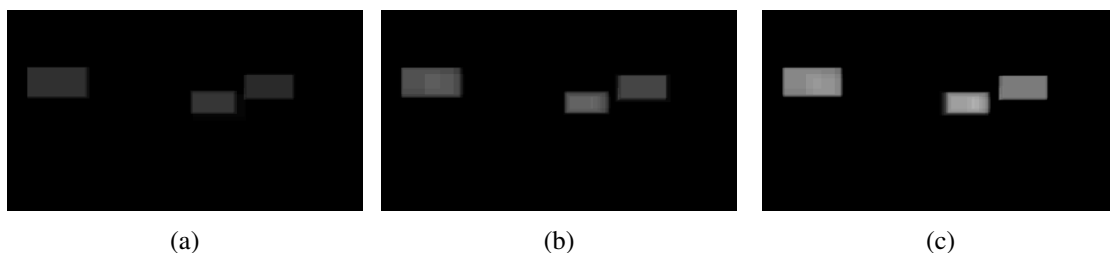


Figura 6.27: *Mapa temporal.*

Como esperado, as deteções consideradas consistentes serão validadas após um determinado período de tempo. Neste caso em particular foram necessários 48 fotogramas para que todas as embarcações fossem consideradas válidas; considerando que o vídeo apresenta 29 *fotogramas/s*, isto significa que as embarcações são validadas em 1,66 segundos, o que é suficiente para um funcionamento correto em tempo-real. Foi definido um limiar, para o qual uma embarcação é considerada válida após o ultrapassar, por tentativa e erro, definido como 100; notou-se que acima desse valor, e dado os valores de incrementação e decrementação associados ao mapa temporal, as validações eram demasiado demoradas e abaixo desse valor resultava em validações incorretas para deteções esporádicas não associadas a embarcações que apresentassem alguma consistência temporal curta.

Na figura 6.28 observa-se uma deteção esporádica, a vermelho, descartada como ruído pelo mapa temporal enquanto mantém, a verde, as deteções válidas.

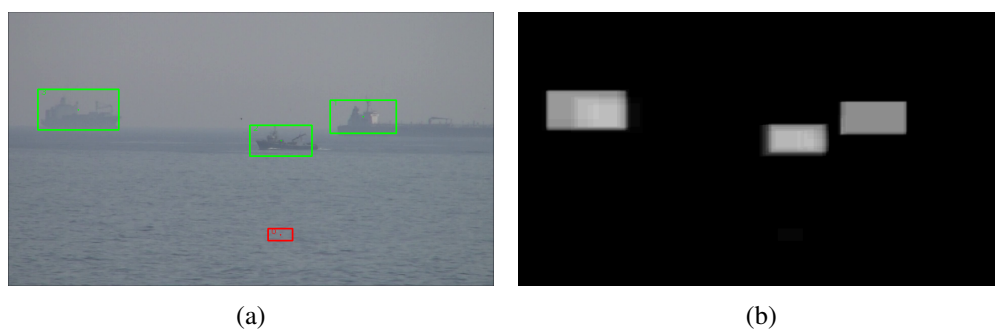


Figura 6.28: *Exemplo de uma detecção errada descartada pelo mapa temporal.*

## Capítulo 7

# Conclusões

### 7.1 Conclusões

Esta dissertação apresenta uma temática muito abordada no mundo da visão computacional - a vídeo-vigilância autónoma de embarcações e estimação das suas posições.

No decorrer deste projeto, realizou-se com sucesso a deteção de qualquer tipo de embarcações e salientou-se a eficácia e robustez do algoritmo desenvolvido para o efeito, podendo facilmente ser adaptado para detetar embarcações autónomas, necessitando, apenas, de um treino diferente do classificador utilizando exemplos positivos de embarcações autónomas.

A validação das embarcações detetadas foi também obtido com sucesso, tendo sido implementado um método bastante simples e de baixa exigência computacional mas com alto desempenho.

Foi conseguido, com sucesso, uma segmentação da imagem eficaz e robusta através da deteção da linha do horizonte e verificou-se a redução da carga computacional imposta na execução do detetor em imagens segmentadas.

Relativamente à estimação da posição no mundo das embarcações detetadas, foi efetuado um estudo aprofundado sobre os conceitos teóricos da geometria projetiva exigidos para a sua implementação.

Foi possível obter leituras das coordenadas GPS da estação de aquisição através do protótipo desenvolvido e fazer uma prova de conceito do mesmo, tendo sido interligados o protótipo da estação de aquisição com o programa de processamento desenvolvido. Foi, também, possível efetuar deteções em tempo real através deste protótipo. No entanto, dada a complexidade e dimensão do trabalho proposto, nem todos os objetivos foram possíveis de realizar, tendo ficado incompleta a implementação do módulo IMU e consequente estimação da orientação da câmara devido, maioritariamente à exigência temporal dos algoritmos de visão. Desta forma, a estimação da posição das embarcações no mundo não foi possível de calcular.

Mais importante, a elaboração desta dissertação serviu para adquirir conhecimentos muito importantes na elaboração de documentos científicos e contribuiu largamente para uma aprendizagem e desenvolvimento metódicos, base de qualquer trabalho com qualidade.

## 7.2 Desenvolvimentos Futuros

Com vista a dar continuidade a este projeto e torná-lo completo e eficaz, são apontados alguns desenvolvimentos futuros interessantes:

- Treino de um novo classificador com exemplos positivos de, apenas, embarcações autónomas, para que a deteção seja parcial;
- Implementação do módulo IMU, leitura das suas medições através do módulo *Arduino* e estimação da orientação da câmara;
- Projeção de um ponto 2D do plano da imagem, referente a uma embarcação, no plano 3D do referencial Mundo;
- Comparação dos dados posicionais 3D calculados com dados obtidos por medição nas embarcações detetadas, para efeitos de cálculo do erro associado.
- Executar o programa desenvolvido em *hardware* acelerado observando a melhoria nos tempos de deteção.

# Referências

- [1] C-Enduro. Asv c-enduro, 2015. Retrieved June 25, 2015, from Planet Ocean Ltd website:[http://planet-ocean.co.uk/wp/?page\\_id=3167](http://planet-ocean.co.uk/wp/?page_id=3167).
- [2] Fast. Feup autonomous sailboat, 2015. Retrieved June 25, 2015, from FEUP website:<http://paginas.fe.up.pt/~jca/fast/media.html>.
- [3] C-Worker. Asv c-worker, 2015. Retrieved June 25, 2015, from Unmanned Systems Technology website:<http://www.unmannedsystemstechnology.com/2014/02/asv-launch-revolutionary-oil-field-services-unmanned-surface-vehicle/asv-c-worker-unmanned-surface-vehicle/>.
- [4] ASV C-Enduro. Asv c-enduro, 2015. Retrieved June 25, 2015, from ThinkDefense website:<http://www.thinkdefence.co.uk/2013/07/autonomous-surface-vehicles-research-contract/>.
- [5] Coast Guard Compass. Coast guard cutter bertholf with suspect vessel, 2015. Retrieved June 26, 2015, from Coast Guard Compass website:<http://coastguard.dodlive.mil/2011/05/from-the-bridge-of-the-bertholf-eastern-pacific-counter-drug-arena/junior/>.
- [6] user Bo47. Railroad tracks, 2013. Retrieved June 27, 2015, from Just Walked By website:<http://www.justwalkedby.com/2013/09/28/railroad-tracks/>.
- [7] Tech CCTV. Pelco ps20 scanner indoor/outdoor up to 35lb 120vac, 2014. Retrieved June 27, 2015, from TechCCTV website:<http://techcctv.com/pelcoptz.htm>.
- [8] Adafruit. Adafruit ultimate gps breakout - 66 channel w/10 hz updates, 2014. Retrieved June 27, 2015, from Adafruit website:<https://www.adafruit.com/products/746>.
- [9] Addictive-Hobby. Ek2-0508 e-sky digital servo, 2009. Retrieved June 27, 2015, from Adafruit website:[http://www.addictive-hobby.com/product\\_info.php?products\\_id=1373&osCsid=05b86218fbebcdc7e1c7f9a4415a05ab](http://www.addictive-hobby.com/product_info.php?products_id=1373&osCsid=05b86218fbebcdc7e1c7f9a4415a05ab).
- [10] F. Moutarde, A. Bargeton, A. Herbin, e L. Chanussot. Robust on-vehicle real-time visual detection of american and european speed limit signs, with a modular traffic signs recognition system. Em *Intelligent Vehicles Symposium, 2007 IEEE*, páginas 1122–1126, June 2007. doi:10.1109/IVS.2007.4290268.
- [11] NguanSoon Chong, YauHee Kho, e MouLingDennis Wong. Visual detection in omnidirectional view sensors. *Signal, Image and Video Processing*, 9(4):923–940, 2015. URL: <http://dx.doi.org/10.1007/s11760-013-0528-0>, doi:10.1007/s11760-013-0528-0.

- [12] M.T. Chan e C. Weed. Vessel detection in video with dynamic maritime background. Em *Applied Imagery Pattern Recognition Workshop (AIPR), 2012 IEEE*, páginas 1–5, Oct 2012. doi:10.1109/AIPR.2012.6528222.
- [13] P. Ramona, L. Nicolas, G. Mercier, H. Guillaume, e R. Garelo. Ship detection in sar medium resolution imagery for maritime surveillance: Algorithm validation using ais data. Em *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, páginas 3690–3693, July 2014. doi:10.1109/IGARSS.2014.6947284.
- [14] B. Ristic. Detecting anomalies from a multitarget tracking output. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(1):798–803, January 2014. doi:10.1109/TAES.2013.130377.
- [15] C. Ruz, C. Pieringer, B. Peralta, I. Lillo, P. Espinace, R. Gonzalez, B. Wendt, D. Mery, e A. Soto. Visual recognition to access and analyze people density and flow patterns in indoor environments. Em *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, páginas 1–8, Jan 2015. doi:10.1109/WACV.2015.8.
- [16] I. Bouchrika e M.S. Nixon. People detection and recognition using gait for automated visual surveillance. Em *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, páginas 576–581, June 2006.
- [17] Q.L. Willard, A.M. Bartlett, L.S. Harrington, e J.C. McKay. A systems approach to autonomous space exploration. Em *Systems and Information Engineering Design Symposium, 2007. SIEDS 2007. IEEE*, páginas 1–5, April 2007. doi:10.1109/SIEDS.2007.4374008.
- [18] S. Wood, T. Allen, S. Kuhn, e J. Caldwell. The development of an autonomous underwater powered glider for deep-sea biological, chemical and physical oceanography. Em *OCEANS 2007 - Europe*, páginas 1–6, June 2007. doi:10.1109/OCEANSE.2007.4302217.
- [19] A. El-Shenawy. The construction of autonomous electric vehicle for land mine detection and localization. Em *Innovative Engineering Systems (ICIES), 2012 First International Conference on*, páginas 91–96, Dec 2012. doi:10.1109/ICIES.2012.6530851.
- [20] A. Subramanian, Xiaojin Gong, J.N. Riggins, D.J. Stilwell, e C.L. Wyatt. Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications. Em *OCEANS 2006*, páginas 1–6, Sept 2006. doi:10.1109/OCEANS.2006.306906.
- [21] C. Kitts, P. Mahacek, T. Adamek, e I. Mas. Experiments in the control and application of automated surface vessel fleets. Em *OCEANS 2011*, páginas 1–7, Sept 2011.
- [22] J. Crook. Deep-sea technology [robotics subsea]. *Engineering Technology*, 5(1):36–39, Jan 2010.
- [23] Y. Lizunkova, T. Hassel, J. Klotz, A. Wolyniec, e F.-W. Bach. Development of filler wire for underwater welding as a repair tool for adaptation on auv. Em *OCEANS 2009 - EUROPE*, páginas 1–6, May 2009. doi:10.1109/OCEANSE.2009.5278190.
- [24] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi, e C. Laschi. The hydronet asv, a small-sized autonomous catamaran for real-time monitoring of water quality: From design to missions at sea. *Oceanic Engineering, IEEE Journal of*, PP(99):1–17, 2014. doi:10.1109/JOE.2014.2359361.



- [25] ASV. C-worker, 2014. Retrieved June 25, 2015, from ASVGlobal website:<http://www.asvglobal.com/oil-and-gas/c-worker>.
- [26] S. Maresca, M. Greco, F. Gini, R. Grasso, S. Coraluppi, e J. Horstmann. Vessel detection and classification: An integrated maritime surveillance system in the tyrrhenian sea. Em *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, páginas 40–45, June 2010. doi:10.1109/CIP.2010.5604209.
- [27] S. Chaojian, X. Kaiyu, P. Jing, e R. Lei. Architecture of vision enhancement system for maritime search and rescue. páginas 12–17, 2008. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-62949112187&partnerID=40&md5=9f1b410377c573a2b3eb73005cf5248c>.
- [28] National Aeronautics And Space Administration. Ultraviolet waves, 2010. Science Mission Directorate. (2010). Retrieved July 18, 2014, from Mission:Science website:[http://missionscience.nasa.gov/ems/10\\_ultravioletwaves.html](http://missionscience.nasa.gov/ems/10_ultravioletwaves.html).
- [29] Domenico Bloisi e Luca Iocchi. Argos – a video surveillance system for boat traffic monitoring in venice, 2009.
- [30] K.D. Ward e R. Tough. Modelling radar sea clutter in littoral environments. Em *Radar, 2008 International Conference on*, páginas 82–87, Sept 2008. doi:10.1109/RADAR.2008.4653896.
- [31] Zhengqiang Jiang, D.Q. Huynh, W. Moran, e S. Challa. Combining background subtraction and temporal persistency in pedestrian detection from static videos. Em *Image Processing (ICIP), 2013 20th IEEE International Conference on*, páginas 4141–4145, Sept 2013. doi:10.1109/ICIP.2013.6738853.
- [32] Wencai Zou, Yao Lu, Mukai Chen, e Feng Lv. Rapid face detection in static video using background subtraction. Em *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on*, páginas 252–255, Nov 2014. doi:10.1109/CIS.2014.146.
- [33] O. Barnich e M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, June 2011. doi:10.1109/TIP.2010.2101613.
- [34] Paul Viola e Michael Jones. Robust real-time object detection. Em *International Journal of Computer Vision*, 2001.
- [35] V. Ablavsky. Background models for tracking objects in water. Em *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, páginas III–125–8 vol.2, Sept 2003. doi:10.1109/ICIP.2003.1247197.
- [36] S. Brusch, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev, e B. van Schie. Ship surveillance with terrasar-x. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(3):1092–1103, March 2011. doi:10.1109/TGRS.2010.2071879.
- [37] B.J. Rhodes, N.A. Bomberger, M. Seibert, e A.M. Waxman. Seecoast: Automated port scene understanding facilitated by normalcy learning. Em *Military Communications Conference, 2006. MILCOM 2006. IEEE*, páginas 1–7, Oct 2006. doi:10.1109/MILCOM.2006.302306.
- [38] Jonathan Guinet Nuno Pires e Elodie Dusch. Asv : An innovative automatic system for maritime surveillance. 2010.

- [39] Rob Wijnhoven, Kris van Rens, Egbert G. T. Jaspers, e Peter H. N. de With. Online Learning for Ship Detection in Maritime Surveillance. Em *Thirty-first Symposium on Information Theory in the Benelux*, páginas 73–80, Maio 2010. URL: <http://vca.ele.tue.nl/publications/data/Wijnhoven2010a.pdf>.
- [40] N. Dalal e B. Triggs. Histograms of oriented gradients for human detection. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 886–893 vol. 1, June 2005. doi:10.1109/CVPR.2005.177.
- [41] Y.M. Wang, Y. Li, e J.B. Zheng. A camera calibration technique based on opencv. Em *Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on*, páginas 403–406, June 2010. doi:10.1109/ICICIS.2010.5534797.
- [42] Byoung-Kwang Kim, Soon-Wook Chung, Moon-Kyu Song, e Woo-Jin Song. Correcting radial lens distortion with advanced outlier elimination. Em *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, páginas 1693–1699, Nov 2010. doi:10.1109/ICALIP.2010.5685158.
- [43] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000. doi:10.1109/34.888718.
- [44] C.X. Guo e S.I. Roumeliotis. Imu-rgbd camera 3d pose estimation and extrinsic calibration: Observability analysis and consistency improvement. Em *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, páginas 2935–2942, May 2013. doi:10.1109/ICRA.2013.6630984.
- [45] F.M. Mirzaei e S.I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *Robotics, IEEE Transactions on*, 24(5):1143–1156, Oct 2008. doi:10.1109/TRO.2008.2004486.
- [46] J. Kellyt e G. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. Em *Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on*, páginas 360–368, Dec 2009. doi:10.1109/CIRA.2009.5423178.
- [47] S. Weiss, M.W. Achtelik, S. Lynen, M. Chli, e R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. Em *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, páginas 957–964, May 2012. doi:10.1109/ICRA.2012.6225147.
- [48] Adafruit. Arduino wiring, 2015. Arduino Wiring. Retrieved June 28, 2015, from Adafruit website:<https://learn.adafruit.com/adafruit-ultimate-gps/arduino-wiring>.
- [49] Rainer Lienhart e Jochen Maydt. An extended set of haar-like features for rapid objection detection. *IEEE ICIP*, página 2002.
- [50] Jerome Friedman, Trevor Hastie, e Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.
- [51] Yoav Freund e Robert E. Schapire. Experiments with a new boosting algorithm, 1996.